Cognitive Networks

Captain Ryan W. Thomas

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Engineering

Dr. Luiz A. DaSilva, Chair
Dr. Y. Thomas Hou
Dr. Allen B. MacKenzie
Dr. Madhav V. Marathe
Dr. Scott F. Midkiff

June 15, 2007
Blacksburg, Virginia

Keywords: cognitive network, cognitive radio, cross-layer optimization, distributed algorithm, game theory, complex networks

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE **15 JUN 2007**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE **Cognitive Networks**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Virginia Polytechnic Institute and State University**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) **Dept of the Air Force AFIT/ENEL WPAFB, OH 45433**	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES

The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**195**	

(ABSTRACT)

For complex computer networks with many tunable parameters and network performance objectives, the task of selecting the ideal network operating state is difficult. To improve the performance of these kinds of networks, this research proposes the idea of the cognitive network. A cognitive network is a network composed of elements that, through learning and reasoning, dynamically adapt to varying network conditions in order to optimize end-to-end performance. In a cognitive network, decisions are made to meet the requirements of the network as a whole, rather than the individual network components.

We examine the cognitive network concept by first providing a definition and then outlining the difference between it and other cognitive and cross-layer technologies. From this definition, we develop a general, three-layer cognitive network framework, based loosely on the framework used for cognitive radio. In this framework, we consider the possibility of a cognitive process consisting of one or more cognitive elements, software agents that operate somewhere between autonomy and cooperation.

To understand how to design a cognitive network within this framework we identify three critical design decisions that affect the performance of the cognitive network: the selfishness of the cognitive elements, their degree of ignorance, and the amount of control they have over the network. To evaluate the impact of these decisions, we created a metric called the price of a feature, defined as the ratio of the network performance with a certain design decision to the performance without the feature.

To further aid in the design of cognitive networks, we identify classes of cognitive networks that are structurally similar to one another. We examined two of these classes: the potential class and the quasi-concave class. Both classes of networks will converge to Nash Equilibrium under selfish behavior and in the quasi-concave class this equilibrium is both Pareto and globally optimal. Furthermore, we found the quasi-concave class has other desirable properties, reacting well to the absence of certain kinds of information and degrading gracefully under reduced network control.

In addition to these analytical, high level contributions, we develop cognitive networks for two open problems in resource management for self-organizing networks, validating and illustrating the cognitive network approach. For the first problem, a cognitive network is shown to increase the lifetime of a wireless multicast route by up to 125%. For this problem, we show that the price of selfishness and control are more significant than the price of ignorance. For the second problem, a cognitive network minimizes the transmission power and spectral impact of a wireless network topology under static and dynamic conditions. The cognitive network, utilizing a distributed, selfish approach, minimizes the maximum power in the topology and reduces (on average) the channel usage to within 12% of the minimum channel assignment. For this problem, we investigate the price of ignorance under dynamic networks and the cost of maintaining knowledge in the network.

Today's computer networking technology will not be able to solve the complex problems that arise from increasingly bandwidth-intensive applications competing for scarce resources. Cognitive networks have the potential to change this trend by adding intelligence to the network. This work introduces the concept and provides a foundation for future investigation and implementation.

Contents

List of Figures

List of Algorithms

Chapter 1

Introduction

Current data networking technology limits a network's ability to adapt to changes and interactions in the network, often resulting in sub-optimal performance. Limited in state, scope and response mechanisms, the network elements (consisting of nodes, protocol layers, policies and behaviors) are unable to make intelligent adaptations. Communication of network state information is stifled by the layered protocol architecture, making individual elements unaware of the network conditions experienced by other elements. Any response that an element may make to network stimuli can only be made inside of its limited scope. The adaptations that are performed are typically reactive, taking place after a problem has occurred. In this dissertation, we advance the idea of *cognitive networks*, which have the promise to remove these limitations by allowing networks to observe, act, and learn in order to optimize their performance.

1.1 What is a Cognitive Network?

In recent years, the words *cognitive* and *smart* have become buzzwords that are applied to many different networking and communications systems. At a minimum, in the current literature we find mention of cognitive radios [1, 2], smart radios [3], smart antennas [4], cognitive packets [5], smart packets [6] and cognitive networks (CNs) [7, 8]. There does not seem to be a common, accepted definition of what these terms mean when applied to a networking technology, let alone an accepted justification for when to add cognition to a network.

The concept of CNs has been bouncing around the collective psyche of the wireless and networking researching world for a while. Mitola [1] makes brief mention of how his cognitive radios (CRs) could interact within the system-level scope of a CN. Saracco [9] refers to CNs in his investigation into the future of information technology. He postulates that the movement of network intelligence from controlling resources to understanding user needs will help "flatten" the network by moving network intelligence further out towards the edges of the network. Mähönen et al. [7] discuss CNs with respect to future mobile Internet Protocol (IP) networks, arguing that the context sensitivity of these networks could have as interesting an application to networks as cognitive radios had to software defined radios. None of these papers, however, express exactly what a CN is, how it should work and what problems it should solve.

1.2 Problem Statement

This highlights the first problem of CNs: identifying their motivation and potential applications. Across the field of computer networking, there is a need to implement network-level objectives in the face of increasing network complexity. Particularly in wireless networks, there has been a trend towards increasingly complicated, heterogeneous and dynamic environments. Although wired networks also have these properties, wireless networks naturally have all three. Due to the changing, shared and mobile nature of the wireless medium, there is a large number of possible inter-radio interactions and operating states. The recent large scale research focus on CR and cross-layer design shows that current networking paradigms are inadequate in dealing with these properties.

This leads to the next open question of CNs: formalizing the research area and its basic operating characteristics. CR and cross-layer design illustrate the pitfalls to approaching a research topic without a clear understanding of the larger system. For instance, CR has no fewer than ten different definitions [10], spanning from simplistic to highly complex. Furthermore, each of these definitions is associated with various different frameworks and architectures. Cross-layer design has similar issues. Although it is common and desirable for researchers to offer differing interpretations on a research area's boundaries, problems arise when contributors are unclear or polymorphous as to exactly what vision they subscribe to. Eventually, the research area becomes muddied, taking on multiple meanings and usages. This can be avoided by clearly defining CNs and characterizing their basic operating framework.

This framework should use direct and indirect observations of the network as an input to a decision making process. The decision making process should use reasoning to determine a set of action choices, implementable in the network parameters. It should ideally use learning so as to be proactive rather than reactive, using past experience to adjust to problems before they occur. Finally, the framework should be extensible and flexible, supporting future improvements, network elements and goals.

The overall challenge for any technology is to meet some need in the best way possible for the least cost. Keeping the first half of this challenge in mind, a CN should provide, over an extended period of time, better end-to-end performance than a non-cognitive network. Cognition can be used to improve the performance of end-to-end objectives such as resource management, Quality of Service (QoS), security, or access control. The limitations of CN applications should come from the adaptability of the underlying network elements and the flexibility of the cognitive process.

In examining the second half of this challenge, the cost of overhead, architecture and operation must justify the end-to-end performance. In almost all cases, implementing a CN will require a system that is more complex than a non-CN. Thus for CNs to be justifiable, the performance improvement must outweigh these additional costs. For certain environments, such as static wired networks with predictable behavior, it may not make sense to convert to cognitive behaviors. Other environments, such as heterogeneous wireless networks, may be ideal candidates for cognition.

In order to address this challenge, a clear measure of CN performance needs to be developed. This measure should incorporate the cost of the design decisions employed by the CN. It should also allow for the creation of design rules that allow network engineers to quickly identify what CN designs decisions are appropriate.

Finally it is necessary to identify applicable problem spaces and re-usable cognitive network solutions. Two types of network problems that exhibit complexity are hard problems and management problems. Although there can be overlap between the two categories, hard problems are characterized as being problems from which there are no "good" solutions. These problems are often characterized as taking non-polynomial time to solve or requiring a tremendous amount of coordination and communication. On the other hand, management problems may be solved in reasonable amount of time, but require excessive or unreasonable levels of human operator intervention. The identification and evaluation of useful CN applications within these two problem areas is needed to speed the acceptance and implementation of the concept.

For the first category, cognition can help determine the amount of effort and communication needed to solve the problem to an adequate degree. In the second category, cognition can be used to imitate the decisions normally made with a human in the loop. Although there is certainly overlap between the two categories, one distinguishing characteristic is that management problems approximate subjective solutions, whereas hard problems approximate objective solutions.

1.3 Methodology

This dissertation defines, develops, classifies, implements and analyzes the CN concept. We promote an approach to achieving, in a distributed manner, end-to-end network objectives in the context of complex, dynamic networks.

In particular, this work formally defines the term cognitive network, differentiating it from other adaptive communication technologies and providing a context and purpose for the concept. In order to further develop this definition, a reference framework that shows the components, interactions and roles of the CN is designed. This framework is inclusive enough to incorporate different objectives, network architectures, hardware, protocol stacks, and cognitive processes. In this manner we design a structural, rather than functional, framework.

To understand the limitations and properties of CNs, an analytical model is developed using a game theoretic framework. Game theory allows for the autonomous, rational operation of various elements within the CN and allows for the separation of individual goals from network objectives. Critical design decisions for the CN are identified from the intersection of this analytical model and a basic architecture for cognition.

These three design decisions – selfishness, ignorance and control – are the central theme of evaluation and analysis of the cognitive network. A metric called the "price of a feature" is specifically developed to evaluate the impact these decisions have on the network goals. This metric also is used to measure the expected and bounded performance of the network under the influence of each design decision.

This analysis gives insight to CN applicability and limitations. It also allows for the classification of different CNs. These CN classes are identified based on the selfish properties of the elements of the network. These classes are used to identify several real-world applications of CNs. Two CN case-studies are constructed from these applications. The first addresses lifetime of wireless multicast routes, while the second addresses lifetime and spectral impact of wireless topologies. Strategies for these problems are developed, and the performance of the network under the design decisions is investigated.

1.4 Thesis Statement

The work in this dissertation represents the first formal investigation of CNs. It provides the first definition, framework, classification, and application for CNs. The definition clearly delineates CNs from other adaptive, cognitive technologies, and creates a distinct set of requirements a system must have to be labeled as such. The framework, inspired by a general three-layer cognitive architecture, articulates the objectives, cognitive process, and network interface. This flexible, distributed framework allows end-to-end objectives to be translated into local, autonomous goals.

In particular, two classes of CNs are identified for problems that utilize a distributed cognitive process consisting of selfish, rational, and distributed elements. For CNs that fall into these classes, we show that they are guaranteed to arrive at a stable Nash Equilibrium (NE) operating point. Furthermore, for one class, this equilibrium is also a Pareto Optimal (PO) point for the elements of the cognitive process. Under the correct alignment with the network objectives, the classes arrive at either a locally or globally optimal state with respect to the end-to-end objectives of the network.

Identified applications of these two classes are incorporated into the CN case-studies. In the first case study, which concerns the multicast route lifetime in a wireless ad-hoc network, the adoption of a CN increases the lifetime of the network up to 125% over non-cognitive approaches. Furthermore, the CN lifetime, under sensible starting conditions, achieves, on average, over 80% of the maximum possible lifetime for the network. We also identify that the critical design decision of control has the greatest effect on the lifetime, more than either selfishness of ignorance.

The second case study, which concerns the transmission power and spectral footprint of a wireless topology, optimally minimizes the maximum transmission power in the network. It also uses less than 12% additional spectrum over the minimum amount for these topologies, which compares favorably against the amount of spectrum required by other approaches in the literature. Furthermore, the CN responds to dynamic changes to the network under various degrees of ignorance, and the optimal amount of knowledge is determined for various degrees of change.

1.5 Outline

This document is structured into seven chapters. Following this chapter, we investigate motivating problems, current solutions and existing models. Chapter 3 consists of the system design, framework and metrics. Chapter 4 identifies two classes of CNs and describes their characteristics. Chapters 5 and 6 delve into the two CN case studies for wireless multicast and spectrum-aware topology control. We close by summarizing the results, drawing conclusions and providing future research directions.

Chapter 2

Related Work

CNs encompass many areas of research, with this work representing the first in-depth research into the topic. This chapter describes related work which both inspires and guides this research. Due to the foundational nature of this research, we begin with a wide orbit, examining the phenomena and technologies that motivate this research, then focusing on how the current solutions address these problems and their contributions. The next section presents a discussion of how to model and describe highly interactive, complex systems. We close with an overview of the recent work from other researchers on this emerging area of research.

2.1 Necessity: The Mother of Invention

In the following section, we investigate three motivating problems that drive research into CNs. They are:

- Complexity: a term used to describe many large, disordered systems of interactions,
- Wireless networking: a rapidly growing area of networking that exhibits many of the features of complexity, and

- QoS: the original motivation for end-to-end network control.

While these areas are hardly a complete list, they are the problems from which the requirements and motivations for CNs were born.

2.1.1 Complexity

A natural place to begin a discussion of the roots of CNs is with complexity. Complexity is a hot area of multi-disciplinary research, spanning diverse and disparate fields. It has found its way into mainstream literature by offering tantalizing parallels between seemingly disjointed areas and providing an underlying structure to seemingly random phenomena [11, 12].

Complexity research has been driven by the failure of traditional reductionist approaches of science to explain behaviors of large, diverse and interconnected systems [13]. Systems of interest to complexity research typically are composed of many interacting parts, each with behaviors simpler than that of the system. Out of these simple individual behaviors, system behaviors that are on the edge between order and chaos are observed. Examples of systems exhibiting various degrees of complexity are economic markets, biological populations, and social networks. Barrett et al. [14] describe these systems as Biological, Information, Social, and Technical (BIST). By examining these systems from a more holistic point of view, complexity attempts to provide an explanation for the behaviors of these systems.

In trying to describe complexity, many characteristics of complex systems have been identified. Size of the system [15] is a commonly cited distinguishing feature. Interaction-based systems (as opposed to algorithm-based systems) are given as another characteristic. Wegner [16] argues that interactions are a more powerful paradigm than algorithms, since algorithms cannot take into account time or the interaction events that occur during computation. For this reason, he claims that interaction-machine behavior cannot be reduced to Turing-machine behavior. Whether or not his postulate is correct, the idea of interaction is a critical aspect to differentiating a complex system.

Another idea used to describe complexity is that complexity is a mix of order and disorder [13]. Rather than being called complex, completely predictable systems are better described as being ordered; completely random systems are better described as chaotic.

Unfortunately, complexity is ill-defined in quantitative terms. There have been several attempts at calculating complexity for specific problem spaces, such as: computational complexity (Landau notation), algorithmic complexity (Kolmogorov) and thermodynamic complexity (entropy). Some generalized quantitative aspects of complexity have been investigated, such as the statistical mechanics of complex graphs [17], but most research has revolved around more qualitative aspects of complexity. Kelly [12] notes that there is no unifying measure of complexity. It is not possible, he argues, to measure the difference in complexity between a Corvette engine and a field of wildflowers. Perhaps the measure of a complex system is a function of the capacity of the observer to understand and decode the source of order in a system.

While complexity can be observed in many different systems, it is certainly present and an issue in networking technology. There are a multitude of possible interactions in a communications network, and yet the only interactions that are typically well-understood and examined are those interactions that are designed and intended. The actual behavior of a network is rarely analytically tractable, meaning that simulation and direct observation are often required to determine system behavior. For instance, interactions between the routing protocol in the network layer and Medium Access Control (MAC) layer require a tremendous amount of statistical analysis and a number of simplifying assumptions to characterize [18].

A great amount of discussion has been spent on two related behaviors of interactions observed in many complex systems: self-organization and emergent behavior. Prehofer [19] defines a self-organized system as one that “is organized without any external or central dedicated control entity.” Self-organization occurs when local elements are able to create a certain structure and functionality [20]. A typical example of self-organization is exhibited by a school of fish, swimming in a school with no visible source of control.

Emergent behavior, on the other hand, is a word used to describe the order that emerges in the system that is greater than the sum of the behavior from the individual components of the system. A common example of an emergent phenomenon is a termite mound, which exhibits remarkable structure from the simple, undirected actions of the termites. Wolf et al. [21] examine the differences between emergence and self-organization, concluding that the two behaviors are not the same. The key difference is the scale at which they occur. Emergence has low level behaviors forming new behaviors at higher levels that are not exhibited at the lower level, and self-organization has a self-reinforcing feedback loop generating structure. In this manner, emergence can exist without self-organization and vice-versa.

Emergence and self-organization are usually observed rather than explicitly designed. One example of an attempt to design emergent properties explicitly is Termite [22], a routing protocol intended to mimic how small insects such as termites or ants find their way from a food source to the colony. It uses digital analogies to pheromones and random variation to mimic swarm intelligence. Even when specifically designed, self-organization and emergence are only apparent in a specific context. For instance, the Termite protocol exhibits signs of self-organization and emergence in routing but none for other behaviors such as power management or QoS requirements.

In general, computer networks exhibit many of the characteristic aspects of complexity – large numbers of highly interconnected, interacting elements and instances of self-organization and emergent behavior. However, many objectives, when faced with complexity, are addressed through the use of human intelligence rather than directly through the network itself.

As a high-level example of this, take the often quoted truism from John Gilmore (in a speech at the Second Conference on Computers, Privacy, and Freedom) that "the Internet treats censorship as a malfunction and routes around it." The Internet he is referring to likely has more than one human mind assisting the technology in routing around any censorship. As an example of this, the great firewall of China successfully censors the routing of HyperText Transfer Protocol (HTTP) over the Internet for over 80 million users. Even with this censorship, some machines are able to route around the firewall, using clever techniques such as proxies, tunneling, and redirects. The network hardware and software does not automatically learn how to avoid this censorship by itself, instead some aspect of "cognition" is needed to route around the censorship.

Complexity is exhibited by systems consisting of a large number of interacting elements. While a complex system may exhibit discernible order through aspects of emergence or self-organization, the system as a whole exists on the edge of order and disorder. Discerning the order in the system has traditionally required human analysis. Particularly for wireless devices, user interaction has often been required to determine optimal operating parameters. Because of their distributed nature, networks of wireless devices need be able to deal with and adapt to complex environments with minimal user interaction. CNs offer this capability for autonomous adaptation and optimization.

2.1.2 Wireless Networks

Complexity can be argued to exist in any network large enough to have a non-trivial amount of interactions. However, a natural place to begin this discussion is with wireless networks. Unlike wired networks, in which data transmitted between nodes on separate wires is isolated from interactions, wireless systems all share a common medium and devices may conflict with any other device in their transmission range. Effectively, there is one physical medium, rather than the many that may exist in a wired network, greatly expanding the number of potential interactions.

Wireless networking technology has become a hotbed of research activity and development in the last decade. With the advent of standards such as IEEE 802.11, Bluetooth, WiMax, CDMA2000 and Universal Mobile Telecommunications System (UMTS), high data rate wireless networks have became real-world systems. However, the next generation of wireless technologies promises levels of complexity well beyond that of the current generation [23].

Ad-hoc networks in particular are an area of great interest for future military and commercial applications. The general idea of ad-hoc networks is that they utilize nodes (i.e. network elements with the ability to transmit and receive Radio Frequency (RF)) as sources, destinations and routers of information. Furthermore, unless infrastructure based wireless networks, ad-hoc networks are highly dynamic, allowing nodes to enter and leave the network at any time. This requires that ad-hoc networks be capable of self-organization to handle the tremendous number of possible interactions.

There have been literally thousands of papers written on various aspects of ad-hoc networks. Most papers turn to simulation when investigating performance because of the difficulty in using other forms of analysis. Unfortunately, it is difficult to predict how even the most basic assumptions used in simulating the network will affect the interactions with other elements in the network. Even well-accepted assumptions as fundamental as how the nodes move [24] or the characteristics of packet traffic [25] can contain unintended consequences. Furthermore, simulation is a poor way of case-checking a protocol, as a stochastic simulation will, in the expected sense, tend to stay out of the fringes of system operation. These limitations to simulation highlight the weaknesses in using it as a design tool. It is critical that network elements be able to adapt to the environment they are actually operating in, rather than the expected environment. Designs should have the flexibility to adapt to current network conditions and environmental inputs.

With the decreasing cost and increasing power of analog to digital converters and computer processors, a new kind of radio that performs signal processing in the digital rather than analog domain has become feasible. These radios, known as Software Defined Radio (SDR), move most of the RF and Intermediate Frequency (IF) functionality, including waveform synthesis, into the digital (rather than the analog) domain, allowing for great flexibility in the modes of radio operation (called personalities). The initial applications for SDR were almost entirely military in nature, and current research is still driven by the needs of military interoperability. The Joint Tactical Radio System (JTRS) is currently the focus of SDR for military applications; it is an attempt to unify cross-service and NATO communications. However, there is increasing interest in using SDR technology in commercial applications.

The flexibility of SDRs also comes with a cost. Whereas hardware defined radios have a fixed number states they can operate in, software defined radios have a practically limitless number of operating states. This increase in state space makes it possible to optimize a radio connection for many different goals; where there once was one possible mode of operation, now there might be many, each with its own strengths and weaknesses. A network of SDRs will only increase the size of the state space. The idea of CR [1] was developed to address the difficulty in determining and achieving the best mode of operation in an SDR.

2.1.3 Quality of Service

There has been a lot of research on how to define a QoS architecture for the Internet, motivated by the desire to provide some sort of end-to-end guarantees by a service provider to its users. This is a difficult area of research because most networking stacks do not operate on an end-to-end paradigm at every level. Even those designed with end-to-end concepts in mind (such as Asynchronous Transfer Mode (ATM), which uses a virtual-circuit based analogy rather than a connectionless one) still have no end-to-end control over the lower two levels of the networks stack. Service Layer Agreements (SLAs) [26], Service Overlay Networks (SONs) [27], and resource reservations [28] made progress in solving these problems for wired networks. Furthermore, QoS specification languages [29] have been developed to translate QoS requirements into actions for the underlying hardware and software.

In wireless networks, however, things are more difficult. In particular, QoS in an ad-hoc network is a difficult problem because of its dynamic and unpredictable nature. Chakrabarti et al. [30] argues that many of the underlying algorithmic problems are most likely NP-complete. Furthermore, they argue that there is a lack of both instantaneous and predictive knowledge to make guarantees of performance. Finally, the size of the network is a limiting factor since large ad-hoc networks require large amounts of computation and communication to propagate network updates. Because of these problems, some authors have taken to using the term *soft QoS* [31] to express the idea of having QoS only during periods of combinatorial stability. Soft QoS differs from traditional QoS in that it only makes guarantees while the network dynamics are slower than the network update function. The concept of soft QoS is indicative of the difficulty that researchers have in solving end-to-end problems in dynamic, complex environments.

Furthermore, QoS has seen a dilution in meaning, with many previously unrelated metrics being including in the definition. This dilution may be related to the fact that while networks operate as collections of connections, for the user (regardless of what application they are using) the only connection that matters is his end-to-end connection. This desire for an end-to-end viewpoint can be seen as desirable for most network (as opposed to network node) operations.

This section introduced three areas that motivate the development of CNs. Complexity was introduced as a characteristic of many large, disordered systems of interactions, requiring a cognitive process to separate the areas of order from disorder. Particularly for wireless networking, which exhibits many of the features of complexity, there is a need for some sort of cognitive system to address the end-to-end requirements of the users. The idea of end-to-end performance is rooted in the study of QoS, but the end-to-end goals of CNs go well beyond those associated with QoS.

2.2 Related Efforts

Having investigated the motivating technology and phenomena for CNs, we now review some existing research areas which are related to the CN concept. We take a look at two areas in particular, CR and cross-layer design. CR adapts radio operation and cross-layer design adapts the network stack; both utilize additional information and control to decide and adopt better-performing behaviors than the underlying technology alone.

2.2.1 Cognitive Radio

CNs have been described by some as networks of CRs [32, 33]. Certainly, the 50% correlation in nomenclature would of itself imply some degree of commonality. This comes from the role that CR (as a model of how to use cognition and adaptation to control the behavior of a complex point-to-point communication channel) had in inspiring the formulation of the CN concept.

CRs were first described by Mitola [33]. Mitola's view of a CR was to build upon the foundation of an SDR, defining it be "a radio that employs model-based reasoning to achieve a specified level of competence in radio-related domains." Haykin [2] defines it as

> "An intelligent wireless communication system that is aware of its surrounding environment (i.e., outside world), and uses the methodology of understanding-by-building to learn from the environment and adapt its internal states to statistical variations in the incoming RF stimuli by making corresponding changes in certain operating parameters (e.g., transmit-power, carrier-frequency, and modulation strategy) in real-time, with two primary objectives in mind: highly reliable communications whenever and wherever needed and efficient utilization of the radio spectrum."

However, Haykin and Mitola have perhaps the most extensive definitions of cognition. Other common definitions, such as those put forth by the IEEE 1900.1 standard [34] and Federal Communications Commission (FCC) [35] propose a more limited concept of cognition – perhaps a more apt name for these definitions would be "adaptive radios."

Cognition as defined by Mitola and as understood by cognitive scientists goes beyond simple adaptation. A common thread between these more advanced definitions of cognition is the idea of a feedback loop. A feedback loop models learning by describing how past interactions with the environment guide current and future interactions. Figure 2.1 illustrates a simple example of a feedback loop first put forward by Col John Boyd, USAF (ret) [36]. Commonly

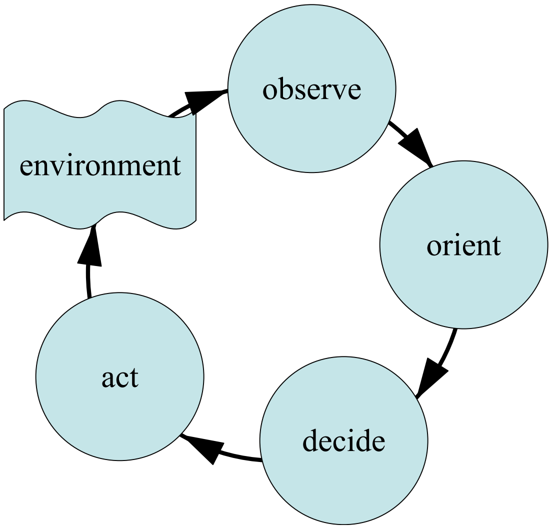


Figure 2.1: The OODA loop, as described by Col John Boyd [36]

called the OODA loop, the model was originally used to help military officers understand the thought processes of their adversaries. However, this loop has been adopted outside of the military, in applications ranging from business management to artificial intelligence. It is remarkably similar to the model Mitola uses to describe the cognition process in CRs. The loop consists of four self-explanatory components, which guide a decision maker through the process of choosing an appropriate action based on input from the environment.

The loop is missing a few important components, however. One is an overarching goal, which should feed in from outside the loop and guide the orientation and decision components by providing a context in which to make a decision. Another missing component is a learning module, which prevents mistakes from previous iterations from being made on future iterations.

Feedback loops such as the OODA loop work because, although the environment in which the decisions are being made may be highly complex, it is not totally random. There is a structure to the complex system that may not be apparent from outside analysis but an attempt to approximate it can be made through iterative cycles of a test-response feedback loop. By simplifying the environment to a black box model, it may be possible to determine some of this structure, particularly if the system is reasonably stationary within the time frame of interest for the adaptations. In a CN, network elements and their interactions are the "black box."

Knowledge in CR is contained within a modeling language such as Radio Knowledge Representation Language (RKRL) [1]. RKRL is an instantiation of Knowledge Query Markup Language (KQML), an interaction language developed for communication between software agents [37]. KQML is itself based on Standard Generalized Markup Language (SGML) and is designed to express and exchange information between intelligent agents and other entities, such as applications or other agents[1]. In addition to exchanging information, KQML is designed to make and respond to requests for information, as well as locate qualified agents.

RKRL, according to Mitola, consists of the following components:

- The mappings between the real world and the various models formed by the cognitive process;
- A syntax defining the statements of the language;
- Models of time, space, entities and communications among entities such as people, places, and things;
- An initial set of knowledge including initial representation sets, definitions, conceptual models, and radio domain models; and
- Mechanisms for modifying and extending RKRL.

KQML, and by association, RKRL, make some demanding assumptions as to the availability and reliability of communication channels. In particular, it is assumed that there are distinguishable connections between all agents, the connections are reliable, and preserve the

[1] It is interesting to note that both eXtensible Markup Language (XML) and HyperText Markup Language (HTML) also have roots in SGML.

order of the messages [38]. These assumptions are appropriate for the local point-to-point scope that CR operates at, but are limiting for a dynamic, complex network of connections.

To attempt to address a language for higher level goals, Mähönen [39] suggests extending RKRL to encompass the the high-level goals of the users of the network. He calls this extension Network Knowledge Representation Language (NKRL) but does not offer any deeper explanation of its structure or operation. Other efforts to encompass higher layer goals for CR include discussion of configuration languages for the End-to-End Reconfigurability Project II (E^2R II) project [40]. E^2R II is focused on end-user re-configurability for a cellular-based paradigm of CR.

Due to the local scope of operation, radios are assumed to be "selfish," operating in a rational manner that benefits the radio, rather than any higher level goals. For this reason, game theory has become an important analysis tool in CR [32, 41, 42]. This model is used to analyze how radios negotiate for parameters which must match for successful link communication. Game theory is discussed more fully in Section 2.3.2.

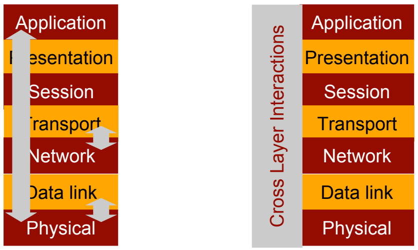
The selfish actions of a CR can typically only be performed on locally-controlled aspects of the network stack. Although Mitola envisioned cross-layer interactions controlling several layers of the network, in most analysis only the physical (PHY) layer and MAC layer are considered. This limits the impacts of changes made by the cognitive process to the radio itself and other radios with which it interacts. Interactions that do not involve radio parameters are not likely to be considered part of CR and instead are usually grouped under the larger heading of cross-layer design.

2.2.2 Cross-layer Design

Cross-layer design violates the traditional layered approach of network architecture design by allowing direct communication or sharing of internal information between nonadjacent or adjacent layers [43]. Particularly for wireless networks, where the interactions between nodes and layers are more pronounced than in wired, cross-layer design offers a tantalizing way to optimize performance. Cross-layer design is controversial because it violates good design principles, leading to possible long-term scalability and reusability issues. Despite being controversial in some corners of the networking community, the perceived benefits of cross-layer design have made it an active area of research.

There are several differences between the architectures of proposed cross-layer designs. Initially, many designs involved simply merging two related layers to accomplish a goal. A common set of merged layers are the PHY and MAC layers. An example of this kind of merging can be found in [44] where the authors combine scheduling with power control to increase throughput through the reduction of medium contention. The advantage to this kind cross-layer design is that for all layers above and below the merged layers, the interfaces and operation of these merged layers appears the same. The disadvantage of this kind of architecture is that it typically optimizes for a single goal, at the expense of other objectives. In the example above, the scheme reduces contention, but it does not explicitly optimize any other areas, such as fairness, node lifetime, or bandwidth allocation. If the goals of the network change, this design will have difficulty adapting to them. Furthermore, if interactions from higher or lower layers are causing behaviors at the joint layer to be sub-optimal, there is little the joint layer can do to prevent this, because the scope of these adaptations is limited to the merged layers.

Another design is to have uni- or bi-directional transfer of information between two non-adjacent layers. This is done by creating new interfaces at the selected layers beyond those used between layers. An example of this kind of architecture can be found in [45], in which information is shared between the transport layer and the PHY layer to increase the throughput and energy efficiency of the network. Adding new interfaces to optimize specific metrics runs the risk of *interface creep*, in which the layered architecture becomes meaningless as designers create and add interfaces without guidance. Furthermore, by opening more interfaces, designers are opening up more interactions, possibly making the network behavior



merged and cross-layer communication architectures

vertical calibration architecture

Figure 2.2: Differences between cross layer architectures that merge or communicate between pairs of layers and vertical calibration.

more complex. Kawadia et al. [46] illustrated this point by showing unintended interactions between the routing protocol and a MAC/PHY cross-layer design.

Recently, cross-layer designs have begun to use a different paradigm to avoid this problem. Instead of building communications between specific layers, proposals such as CrossTalk [47], ECLAIR [48], CLD [49] and the architecture of Wong [50] utilize a parallel structure that acts as a shared database of the system state accessible to whichever layers choose to utilize it. This difference is illustrated in Figure 2.2. Srivastava [43] calls these kinds of architectures *vertical calibrations* because the system is jointly tuning several parameters over the whole stack to achieve an application-level objective.

The advantage to these vertical calibration architectures is that they provide a structured method for accessing parameters, controlling, and sensing the status of each layer. This solves some of the concerns about cross-layer design decreasing the utility of the layered architecture. To its disadvantage, even with its larger scope the cross-layer interactions are stack-centric and do not incorporate nodes outside the scope of the stack. For instance, PHY layer properties (battery life, bandwidth usage) of nodes not within control of a MAC layer are ignored in a MAC/PHY cross-layer protocol. Inter-nodal communication is limited to the scope of the layers being cross-designed.

Cross-layer designs still have problems supporting trade-offs between multiple goals and the effect of these increased number of interactions in achieving those goals. While vertical calibration architectures can support multiple cross-layer optimizations, there is little discussion of how to de-conflict these optimizations. Cross-layer designs perform independent optimizations which do not account for the set of performance goals as a whole. Trying to achieve each goal independently is likely to be sub-optimal, and as the number of cross-layer designs within a node grows, conflicts between the independent adaptations may lead to adaptation loops [46]. Adaptation loops occur when conflicting goals cause a system to fail to converge to a stable operating point.

Cross-layer designs are memoryless adaptations which will respond the same way when presented with the same set of inputs, regardless of how poorly the adaptation may have performed in the past. The ability to learn from past behavior is particularly important in light of the fact that understanding the interactions between layers is difficult. Several protocol architectures incorporate adaptation, but often no discussion is given to intelligence, learning, or pro-active adaptation.

Cross-layer design offers great performance promises through the increased flow of information between layers and the additional control this information offers. Beyond its weaknesses in multiple-objective optimization, learning and scope, care must be taken to consider possible undesirable (and unpredictable) interactions across parameters in the various layers.

2.3 System Models

A deeper understanding of the complex system behavior of a network can come from having a set of models that allow investigation into their properties. These models must be general enough to be applied to many different system implementations and problems, but specific enough to capture the causes of the behavior, without abstracting key aspects. Varian [51] advises us to "write down the simplest possible model you can think of, and see if it still exhibits some interesting behavior. If it does, then make it even simpler." With this advice in mind, we present four models that describe four different aspects of behavior. Multi-agent systems describe the model used for the design of a system of distributed, intelligent agents. Game theory is a model used to understand the behavior of systems consisting of cooperative and non-cooperative rational agents. Interaction models describe the state transitions and fixed points of complex systems of many interacting elements. Finally, machine problem solving contains a wide-ranging set of techniques for determining the optimal solutions for large problem spaces.

2.3.1 Multi-Agent Systems

Multi-Agent Systems (MASs) are a catch-all model that includes variations on systems consisting of multiple software agents. MASs are a direct descendant of Distributed Artificial Intelligence (DAI) [52], which pioneered research into such questions as task allocation, coordination, cooperation and interaction languages between agents. MAS research goals are even broader than the artificial intelligence goals of DAI, consisting of almost any system with a distributed network of software agents.

Although MASs have no universally accepted definition, we will adopt the definition in Jennings [53], which relies on three concepts: *situated*, *autonomous*, and *flexible*. "Situated" means that agents are capable of sensing and acting upon their environment. The agent is generally assumed to have incomplete knowledge, partial control of the environment, or both limitations [54]. "Autonomous" means that agents have the freedom to act independently of humans or other agents though there may be some constraints on the degree of autonomy each agent has. "Flexible" means that agents' responses to environmental changes are timely and pro-active and that agents interact with each other and possibly humans as well in order to solve problems and assist other agents.

In [55], Dietterich describes a standard agent model consisting of four primary components: observations, actions, an inference engine, and a knowledge base. In this agent model, reasoning and learning are a result of the combined operation of the inference engine and the knowledge base. By our definition, reasoning is the immediate process by which the inference engine gathers relevant information from the knowledge base and sensory inputs (observations) and decides on a set of actions. Learning is the longer term process by which the inference engine evaluates relationships, such as between past actions and current observations or between different concurrent observations, and converts this to knowledge to be stored in the knowledge base. This model fits well within most of the cognitive architectures previously mentioned, and we shall use it as our standard reference for our discussion of distributed learning and reasoning.

According to Wooldridge [56], there are two aspects of design that drive MAS research:

- Agent design: the task of creating software agents that are able to carry out tasks autonomously;
- Society design: the task of creating software agents that interact in a manner to carry out tasks in an uncertain environment.

The idea of self-organization is implicit in society design; furthermore software agents are designed to operate in large, distributed environments. For these reasons, MASs have several appropriate properties for describing CN behavior: they can address complex systems, exhibit varying degrees of machine learning, have a naturally distributed nature, and can operate in less-than-reliable environments.

Since MASs are also interaction based, there have been some efforts to use emergent and self-organizing properties of MAS to address complex systems. Wolf [57] examines a software engineering design process for creating self-organizing, emergent systems. The technique addresses complexity by having agents achieve desired macroscopic goals from local interactions – without understanding the underlying interactions that define how a system behaves. Cooperative multiagent agent systems, called Cooperative Distributed Problem Solving (CDPS), also possesses inherent group coherence [58]. In other words, motivation for the agents to work together is inherent in the system design. For a more detailed investigation of MAS and how CNs can be framed as such, we refer to the book chapter by Friend, Thomas, MacKenzie and DaSilva [59], entitled "Distributed Learning and Reasoning in Cognitive Networks: Methods and Design Decisions."

2.3.2 Game Theory

Game theory models the behavior of a system as a game played by multiple rational players. Game theory requires that each player have an *action space* of possible actions and a *utility function*, which represents the relative desirability of of a player's action (chosen from his action space) in combination with actions from the rest of the players (chosen from their action space). Players are said to play *rationally* if they try to choose an action that, in conjunction with the other player actions, maximizes their utility function.

Although game theory is widely used, due to the variety of game models and author preferences there are several variations of notation. In this work, we will primarily use a strategic non-cooperative game $\Gamma = \langle N, A, u \rangle$ consisting of components:

1. Player set $N : N = \{1, 2, \ldots, n\}$ where n is the number of players in the game.

2. Action set $A : \mathbf{a} \in A = \times_{i=1}^{n} A_i$ is the space of all action vectors (tuple), where each component, a_i, of the vector $\mathbf{a}$ belongs to the set A_i, the set of actions of player i. Often we denote action profile $\mathbf{a} = (a_i, a_{-i})$ where a_i is player i's action and a_{-i} denotes the actions of the other $n-1$ players. Similarly, $A_{-i} = \times_{j \neq i} A_j$ is used to denote the set of action profiles for all players except i. In this repeated game, we will denote the ordering of the decisions by a superscript. Thus player i decides to play an action a_i^t at stage t. A mechanism of choosing a series of action choices by an element is called a *strategy*.

3. Utility u: For each player $i \in N$, utility function $u_i : A \rightarrow \mathbb{R}$, models players' preferences over action profiles. $u = (u_1, \ldots, u_n) : A \rightarrow \mathbb{R}^n$ denotes the vector of such utility functions; higher values represent improved attainment of the preference.

Several properties of certain action vectors have been identified. Perhaps the most famous is the NE. The NE is an action vector that corresponds to the mutual best response for all players. In other words, at NE, no individual player can benefit from unilateral deviation.

Definition 2.3.1 (Nash Equilibrium)**.** *An action vector* $\hat{\mathbf{a}}$ *is a NE if, for every player* i *and every action vector* $\mathbf{a}$

$$u_i(\hat{a}_i, \hat{a}_{-i}) \geq u_i(a_i, \hat{a}_{-i}) \tag{2.1}$$

An action vector is said to be Pareto Optimal (PO) if there is no other outcome that makes every player at least as well off while making at least one player better off.

Definition 2.3.2 (Pareto Optimal)**.** *An action vector* $\mathbf{a}'$ *is PO if there does not exists an action vector* $\mathbf{a}$ *such that*

$$u_i(\mathbf{a}) \geq u_i(\mathbf{a}') \quad \forall i \tag{2.2}$$

with a strict inequality for at least one element i*.*

Game theory was originally used by economists, but has become a useful tool for analyzing autonomous systems in computer science and engineering. In particular, ad-hoc networks [41] and CRs [32] have received game theoretical analysis. Game theory is useful in determining rational fixed points (the set of system states from which the system is static, with no elements changing state), but only for systems that can be modeled as containing competing elements.

A useful application of game theoretical analysis to a network-based problem can be found in Roughgarden's work [60] on bounding flow cost. Here, a rational game between competing flows is used to determine the *price of anarchy* [61] for a routing game. The price of anarchy is defined as the worst-case performance difference between players following selfish, rational choices and those acting optimally following a socially-conscious and communal mode of operation. Roughgarden's results show that in some constrained cases, it is possible to put a fixed bound on the price of anarchy. By recognizing that the NE flow is a rational fixed point, this technique can be used to determine a bound on the cost of selfish behavior.

There have been several interesting applications of game theory beyond pure analysis. An entire sub-genre of game theory called mechanism design [62] deals with the design of games to assure the achievement of desired outcomes. Related to this idea are incentives, which change a player's utility through the use of a system of payments from one player to another. Incentive schemes have been used for QoS [63], cooperation in ad-hoc networks [64, 65, 66], and trust [67], to make a short list.

2.3.3 Network Interactions

In order to study the effects of interactions in complex systems there exists a large family of abstract models of networks of interactions. Network interaction models consist of frameworks that describe the way that elements and their actions affect one another.

Perhaps the earliest model of interactions is the Ising model of ferro-magnetic phase transitions, examined in depth by Glauser [68]. The Ising model is part of a set of related models called *infinite particle systems*, of which the voter model [69] and the contact model [70] are mathematical representations. Infinite particle systems are infinite, d-dimensional lattices whose vertices can take on one of two states. These states propagate through the lattice at a rate dependent on the states of neighboring vertices. While the states are discrete (binary), the time mapping in these systems is continuous.

The influence model [71] is a discrete version of the infinite particle system, with arbitrary graph structure and weighting. It is constructed as a discrete time Markov process with transition probabilities affected by the neighboring elements. The influence model has been used to analyze a variety of behaviors in complex systems such as the spread of fads, failures in electrical grids, and epidemics. In this manner, this model has already demonstrated some practical use.

Simple Petri Nets are a means of modeling systems that contain concurrency, serializability, synchronization or resource sharing aspects [72]. Simple Petri Nets are effectively an extension of finite state machines, and they carry with them the flexibility and generality of this model. When the time between state transitions is assumed to be a Markovian process, there is overlap between the results of queuing theory and Petri Nets. This type of Petri Net is called a stochastic Petri Net. Stochastic Petri nets are useful for modeling interactions in processes that share resources, such as micro-processors or TDMA time slots.

Another model used to examine complex systems is Cellular Automata (CA) [73], which are infinite, regular grids of cells, each in one of a finite number of states. The grid can be in any finite number of dimensions. The most commonly studied CA are represented by 1-dimensional rectangular graphs of black and white boxes. Each box represents an automata, with the color representing its binary state. The future state of an automata is a function of the neighboring states. There are 256 possible distinct "rules" to govern behavior in the 1-dimensional case. Cook [74] shows that CA rule 110 is equivalent to a Turing machine, capable of universal computation. Clearly, for such a simple model, CA are capable of representing complex behaviors.

Barrett et al. introduce a general model for investigating BIST systems in a series of papers [75, 76, 77]. The term given to this model is Sequential Dynamic System (SDS). SDSs are related closely to CA. Whereas CA are dimensional, their interactions "wired" to only neighboring nodes, SDSs are multidimensional, potentially "wired" to any node, allowing interactions between any element in the network. Another difference is that most CA compute their transition functions synchronously, whereas SDS compute the states sequentially.

A common simplification of the SDS involves limiting transforming the network Boolean states. This simplification means that the transition functions can be defined as logical operations. This subclass of a the SDS is called a boolean network [11, 78]. A further subclass of the SDS is the Synchronous Dynamic System (SyDS) in which the order of local transition function evaluation (π) is eliminated in favor of synchronous evaluation.

A simplification of the boolean SyDS is called the Kauffman model. The Kauffman model consists of nodes with some average connectivity. The state of the node in the next time step is modeled by a probability function; the closer that probability is to 0.5, the more often the state will change given a different set of input states from its neighbors. Kauffman models were originally designed to simplify the complex modeling of gene-regulatory networks [11]. In these networks, proteins regulate their own production and the production of other proteins through simple computational operations. Through subtle environmental cues, feedback loops, amplification, degradation and inter-process communication, regulatory networks are able to obtain global objectives through local behaviors [79, 80, 81].

2.3.4 Multiple Objective Optimization

Most engineering problems require tradeoffs to determine how to juggle multiple objectives. When a problem has multiple objectives it will not be able to optimize all metrics indefinitely, eventually reaching a point in which one metric cannot be optimized without affecting another. This area of optimization research is called Multiple Objective Optimization (MOO). MOO can be defined as the problem of finding [82]

> ...a vector of decision variables (*actions*) which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term "optimize" means finding such a solution which would give the values of all the objective functions acceptable to the designer.

A more formal definition comes from [83]:

Definition 2.3.3 (Multiple Objective Optimization). *A multi-objective optimization attempts to find the vector of actions* $\mathbf{a}^* = (a_1^*, a_2^*, \ldots, a_n^*)$ *that exists in the feasible region* F *defined by the* m *inequality constraints* $g_i(\mathbf{a}) \geq 0 \quad i = 1, 2, \ldots, m$ *and the* p *equality*

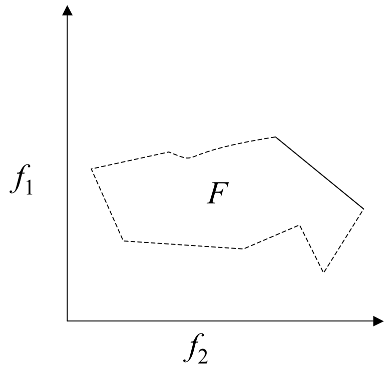


Figure 2.3: PO front for two objective optimization. F represents the feasible region that meets the inequality constraints. The solid line is the Pareto front when we wish to simultaneously maximize f_1 and f_2.

constraints $h_i(\mathbf{a}) = 0 \quad i = 1, 2, \ldots, p$ *to optimize the vector function*

$$\mathbf{f}(\mathbf{a}) = (f_1(\mathbf{a}), f_2(\mathbf{a}), \ldots, f_k(\mathbf{a}))$$

where $\mathbf{a}$ *is the vector of actions resulting from system decisions,* $g_i(\mathbf{a})$ *and* $h_i(\mathbf{a})$ *represents constraints on the solution, and the vector* $\mathbf{f}(\mathbf{a})$ *represents the* k*-objective criteria that must be optimized.*

The definition of "optimal" for the vector function $\mathbf{f}(\mathbf{a})$ is unclear. We assume that it means maximize each element in $\mathbf{f}$ for simplicity, although it could be considered in terms of maximization or some other mixed form. A PO front [83] is derived from these local optimizations to describe the set of actions from which no goal can be improved without worsening another (see Definition 2.3.2). This is illustrated in Figure 2.3.

Finding or even approximating the PO front is a difficult task. One reason is that the vector function of $\mathbf{f}(\mathbf{a})$ is often *noncommensurable*, meaning that the individual functions are in differing units. Thus transforming the problem into a single objective optimization through mathematical operations is a limiting manner of solving the problem – it only finds a single point in the front. This group of approaches, called aggregating, is illustrated by the common weighting approach. To achieve meaningful results, weighting requires a knowledge of the range, relative importance, unit relationship and behavior of each objective function. This is because the resultant decision vector $\mathbf{a}^*$ will only be a subset of points in the Pareto front, determined and dependent on these assumptions.

For this reason, simply consolidating a MOO to a single objective optimization is an undesirable method of solving a problem without a priori understanding of the system. Take, as an example, the wireless networking problem of maximizing both transmission power and throughput. Assume that there are out of all the operating points that meet the designer's requirements (falling into F), three make up the Pareto front. They are A, with a (power^{-1}, bandwidth) of (1 dBm^{-}1, 10 Mb/sec); B, with (0.2 dBm^{-1}, 20 Mb/sec); and C, with (0.1 dBm^{-1}, 30 Mb/sec). Knowing the range, magnitude and behavior of the Pareto front values in advance would allow the scaling, weighting and comparison of the purchaser's utilities for each of the cars. However, these values are not known in advance, and attempting to perform the weighting without this information will likely result in a PO solution that is suboptimal to the purchaser's desires.

2.3.5 Machine Problem Solving

In the following paragraphs we give a brief discussion of possible mechanisms for problem solving in a machine environment. There are many other possibilities, including combinations or hybridization of the following strategies; the choice of algorithm depends on what the network goals are, how these problems are set up, and what the physical structure of the network allows. Complex cognitive processes may have several problem solving processes operating simultaneously, each using mechanisms appropriate for the problem being solved.

We focus our attention towards machine learning, since we assume that the complex and dynamical nature of the network problem space precludes more static methods such as linear programming or solving systems of differential equations. One common method cited when discussing machine learning and artificial intelligence is the area of neural networks.

Neural networks use a bottom-up method of learning, simulating the biological neurons and pathways that the brain is believed to use. A series of these artificial neurons analyze different aspects of known inputs with some amount of unknown corruption. Pattern recognition is a common and straightforward application of neural networks. If network responses are modeled as a noisy pattern, a neural network could be used to categorize the pattern into predetermined responses.

Genetic Algorithms (GAs) and Evolutionary Algorithms (EAs) are used to optimize over large solution spaces where exhaustively searching would be too costly. By imitating the process of evolution (selection, recombination, and mutation), genetic algorithms are able to explore these large solution spaces for local optima. Genetic algorithms have many applications, but work best for centralized problems where the environment is well known. For this reason, if most of the current network state is known, genetic algorithms could be used to determine optimal behaviors.

Recently, artificial intelligence has focused on expert systems [84]. Expert systems are often used to make decisions in a narrow field of knowledge (a common application is medical diagnosis) and imitate the "intuition" that a professional in the field would have in finding a solution by asking a series of questions (typically with only "yes," "no," and "no response" answers). Expert systems can be useful in determining how to prioritize goals and solve problems that have a limited number of possible inputs.

Kalman filters [85] contain an adaptive algorithm for feedback control. Usually found in systems that contain significant amounts of system noise, the Kalman filter is a recursive filter used to estimate the actual and future state of the system based on noisy Gaussian measurements. The ability to act as a dynamic filter means a Kalman filter can be useful for tracking and maintaining a particular performance metric in a changing or noisy system.

Learning automata [86] are a distributed, adaptive control solution to identify the characteristics of an unknown feedback system. Learning automata maintain a probabilistic function for deciding what action to make. The function converges to decisions that generate desired responses in the system. Typical applications for learning automata include problems where there are many dynamic elements interacting with a complex system, such as intelligent vehicles [87], cross-layer optimization problems [88], or routing problems [86]. If the problem is distributed and requires little state information, learning automata can be a good approach.

For a more complete investigation of machine learning techniques (which goes beyond the scope of the research here), we again refer book chapter by Friend et al. [59].

When dealing with MOO, the task of finding the Pareto front is a more difficult problem than just finding a single optimal point. However, there have been several approaches proposed in literature to investigate this. Coello Coello [83] provides a survey of various EAs used to find the set. In real world systems, GAs have been proposed for use in controlling CRs [89], but the distributed nature of CNs requires a slightly different architecture. One solution is parallel Multiple Objective Optimization Evolutionary Algorithm (pMOOEA) over multiple devices in the network, which is investigated in [90].

The models presented in this section give a deeper understanding of complex system behavior and provide insight into the design of processes that interact with them. MASs model the architecture and behavior of systems of distributed, intelligent agents. To understand the fixed point behavior of systems consisting of cooperative and non-cooperative rational agents, game theory can be used. Interaction models capture the state transitions that govern the operation of a complex system of many interacting elements. The different models of problem solving describe how the state transitions of the system are arrived at and what properties a system should have to be successfully solved.

2.4 Related Cognitive Network Work

The work presented in this dissertation started with our first paper at DySPAN [91] which was the first formal investigation into the field of CNs. However, since this first publication, the concept has become a burgeoning area of research.

Recent research can be divided into two categories: Cognitive Radio Networks (CRNs) and CNs. In the first category, we begin with work from Mitola and his original thesis on CR. Here he mentions how CRs could interact within the system-level scope of a CN [1]. Neel continues this line of thinking in [92], where he investigates modeling networks of CRs as a large, multiplayer game to determine convergent conditions. This kind of thinking is also observed in Haykin's paper on CR [2], where he examines multi-user networks of CRs as a game.

The focus of CRNs, as with CRs, is primarily on MAC and PHY layer issues, but now operating with some end-to-end objective. In a CRN, the individual radios still make most of the cognitive decisions, although they may act in a cooperative manner. Currently suggested applications for CRNs include cooperative spectrum sensing [93, 94] and emergency radio networks [95]. From a more general perspective, Raychaudhuri [96] presents an architecture for CRNs.

Perhaps the first mention of a CNs rather than a CRN comes from Clark [97]. Clark proposes a network that can

> "assemble itself given high level instructions, reassemble itself as requirements change, automatically discover when something goes wrong, and automatically fix a detected problem or explain why it cannot do so."

This would be accomplished with the use of a Knowledge Plane (KP) that transcends layers and domains to make cognitive decisions about the network. The KP will add intelligence and weight to the edges of the network, and context sensitivity to its core. Saracco also observed these trends in his investigation into the future of information technology [9], postulating that the change from network intelligence controlling resources to having context sensitivity will help "flatten" the network by moving network intelligence into the core and control further out to the edges of the network.

CRNs differ from CNs in that their action space extends beyond the MAC and PHY layers and the network may consist of more than just wireless devices. Furthermore, CN nodes may be less autonomous than a CRN node, with the network elements cooperating to achieve goals, a centralized cognitive process or a parallelized process that runs across several of the network elements.

Mähönen discusses CNs in the context of future IP networks and cognitive trends in a series of papers. In his earliest paper, he discusses CNs with respect to future mobile IP networks, arguing that the context sensitivity of these networks could have as interesting an application as CRs [7]. He then examines CNs as part of a larger paper on cognitive trends [39]. He discusses how CRs may be just a logical subset of CNs. He also brings up the idea of NKRL to express and communicate high level goals and policies.

More recently, several research groups have proposed CN-like architectures. These architectures can be categorized into two objectives: the first centers on using cognition to aid in the operation and maintenance of the network, while the second centers on cognition to solve "hard" problems, problems that do not have a feasible solution other than the use of cognition.

Falling into the first category, the E^2R II [98] is designing an architecture that will allow the seamless reconfiguration of a network in order to allow for universal end-to-end connectivity. Although E^2R II is an ambitious project with many facets, the overarching goal is one of maintaining connectivity to the user. This is similar to the goal of the m@ANGEL platform [99], which attempts to provide an CN-like architecture for mobility management in a heterogeneous network. Both of these architectures are focused on the operation and maintenance of 4G cellular and wireless networks.

In contrast, the Centre for Telecommunications Value-Chain Research (CTVR) at Trinity College [100] has presented a proposal for a CN platform that consists of reconfigurable wireless nodes. Although focused on wireless operation, these nodes are able to solve a variety of problems by modifying or changing the network stack based on observed network behaviors. The possible objectives of these networks can extend beyond mobility management and connectivity. Similar to the CTVR work but less dependent on the wireless focus, Mähönen proposes a general architecture utilizing a collaborative Cognitive Resource Manager (CRM) that provides cognitive behavior from a toolbox of machine learning tools such as neural networks, clustering, coloring, genetic algorithms, and simulated annealing. The work this chapter describes also falls under this objective, attempting to provide a general cognitive architecture capable of solving a variety of hard problems, rather than being tied to network operation issues.

2.5 Summary

In this chapter we have established that complexity is a real problem facing network operations and management. In attempting to maintain end-to-end goals, current solutions are limited in both scope and methodology. There is a need for a new framework for network operations. To analyze and build this new framework, a set of models was presented that can be used to examine the operations, behaviors and interactions in a complex system.

Chapter 3

Cognitive Network Design and Analysis

This chapter defines the specific the solution space occupied by CNs, illustrated by a simple example. This definition is used as the foundation for our CN framework, a layered approach that supports translating end-to-end goals into the modifiable elements of the network via a cognitive process. The operation and cognitive functionality of each layer in the framework is described. At each layer of the framework, we identify a critical design decision, a design tradeoff that significantly affects the operation and performance of the cognitive network. To quantify the impact of these design decisions, we develop a special metric: the price of a feature.

3.1 System Overview

In this section we define CNs and introduce a simple example to illustrate the concept. A clear definition is an important starting point for this research as it sets limits to the concept, bounding the applications and implementations that can be attributed to it. It also differentiates CNs from other technologies, which creates a starting point for comparing and contrasting it with other approaches.

3.1.1 Definition

The following is the first definition of a CN in the literature. It was first presented at the IEEE DySPAN conference in 2005 [91].

> A cognitive network is a network with a cognitive process that can perceive current network conditions, and then plan, decide and act on those conditions. The network can learn from these adaptations and use them to make future decisions, all while taking into account end-to-end goals.

The cognitive aspect of this definition is similar to that used to describe CR and broadly encompasses many simple models of cognition and learning. The ability for a network to use current network conditions to adapt itself echoes how cross-layer design uses information from one layer of the network to inform and guide actions in other layers.

However, CNs are distinguished from both of these technologies by their end-to-end scope. Without this scope, the system is perhaps a CR or cross-layer adaptation, but not a CN. *End-to-end*, in this definition, denotes all the network elements involved in the transmission of a data flow. For a unicast transmission, this might include everything from subnets, routers, switches and virtual connections to encryption schemes, mediums, interfaces and waveforms. CRs and cross-layer design have only a local, single element scope related to their functional technology – the physical medium or the protocol layers being adapted.

3.1.2 A Simple Example

This example and its description are inspired and influenced by Daniel Friend's example in our IEEE Communications Magazine article [101].

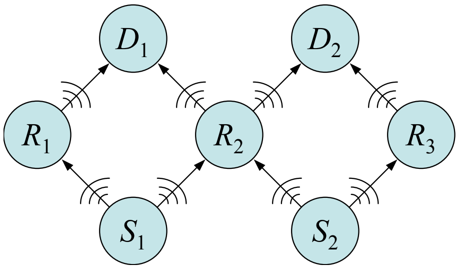


Figure 3.1: Simple relay network for a wireless network. Vertices represent wireless connectivity.

To illustrate the need for end-to-end rather than link adaptations, consider an ad-hoc data session between a source node S_1 and a destination node D_1 as shown in Figure 3.1. The source node does not have enough power to reach D_1 directly, so it must must route traffic through intermediate nodes R_1 and/or R_2. Assume that the end-to-end goal is to have the highest probability of successful transmission. The routing layer will determine routes based on minimum hop count which, in this case, includes either R_1 or R_2. Node S_1 will make a link-layer adaptation, selecting between R_1 and R_2 based on their Signal to Interference and Noise Ratio (SINR). From the standpoint of the link layer in node S_1, this ratio correlates with the probability that the transmitted packets will arrive correctly at the relay node. However, without additional information, this selection does not guarantee anything about the end-to-end packet delivery probability from S_1 to D_1.

In contrast to a link adaptation, the CN might use some combination of observations from all nodes to compute the total path outage probabilities from S_1 to D_1 through R_1 and R_2. This shows the benefit of an end-to-end scope, but there is another advantage to the CN, its cognitive capability. To illustrate this, we modify the original scenario to include both S_1 and S_2 as source nodes, both routing traffic through R_2. Suppose that the learning mechanism measures outages by determining the fraction of packets successfully delivered from the source to its destination.

If R_2 becomes congested because of a large volume of traffic coming from S_2 this becomes apparent to the cognitive process by the lack of successful packet delivery statistics provided to S_1 and S_2. The learning mechanism recognizes that the system has changed and that routing through R_2 is not optimal. The cognitive process then directs the traffic toward another route. The CN does not explicitly know that there is congestion at node R_2 because this information is not included in the SINR observations. Nevertheless, it is able to infer from the reduced throughput that there may be a problem. It is then able to respond to the congestion, perhaps by routing traffic through R_1 and/or R_3. This example shows the power of CNs in optimizing end-to-end performance as well as reacting to unforeseen circumstances.

3.2 System Framework

In this section, we present the system framework for a CN. There is a danger in trying to create a framework that is too specific to any problem, as this often is restrictive for both application and implementation. Successful communications frameworks, such as the Open Systems Interconnection (OSI) stack, have benefited from generality by allowing a diverse range of applications (e.g. voice, text, data, video) and implementations (e.g. TCP or UDP, wireless or wired, IP or ATM). In designing the system framework for a CN, the same principles were applied to allow the engineer to design a solution that solves a particular problem while staying inside the limits of the framework.

The three-tier model of cognition motivates our selection of design decisions. This model was first suggested by David Marr [102] in his work on computer vision and then generalized for other cognitive tasks in [103]. It describes the general functionality required of a cognitive entity. The model consists of the behavioral layer, the layer of cognition that dictates what motivates the decisions of the elements; the computational layer, which decides on the course of action for an element; and the neuro-physical layer, which enacts these decisions into the environment.

From this concept, we draw a three-layer framework with each layer corresponding roughly to the layers in the cognitive model. At the top layer are the goals of the elements in the network that define the behavior of the system. These goals feed into the cognitive process, which computes the actions for the system to take. The Software Adaptable Network (SAN) is the physical control of the system, providing the action space for the cognitive process. This framework is illustrated in Figure 3.2.

In our framework, we consider a cognitive network broken into one or more cognitive elements, operating in some degree between autonomy and full cooperation. If there is a single cognitive element, it may be physically distributed over one or more devices in the network. If there are multiple elements, they may be distributed over a subset of the devices in the network, on every device in the network, or several cognitive elements may reside on a single device. In this respect, the cognitive elements can operate in a manner similar to a software agent in a MAS. This wide variety of cognitive process architectures is intended to give the designer the freedom to create the best cognitive process against the tradeoffs of autonomy and aggregation, centralization and distribution, simple and complex processes, selfishness and altruism, and single and multiple goals.

3.2.1 Requirements Layer

The top level component of the CN framework includes the end-to-end goals, Cognitive Specification Language (CSL), and the resultant cognitive element goals. The end-to-end goals, which drive the behavior of the entire system, are put forth by the network users, applications or resources.

The end-to-end goals are critical to the CN concept because they provide the proper scope for its behavior. This scope clearly delineates CNs from CRs and cross-layer design. Goals in a CN are based on end-to-end network performance, whereas CR goals are based on local performance. Furthermore, CN goals are pulled from the users, applications, and resources of the network, whereas CR goals are pulled from the users, applications and resources of

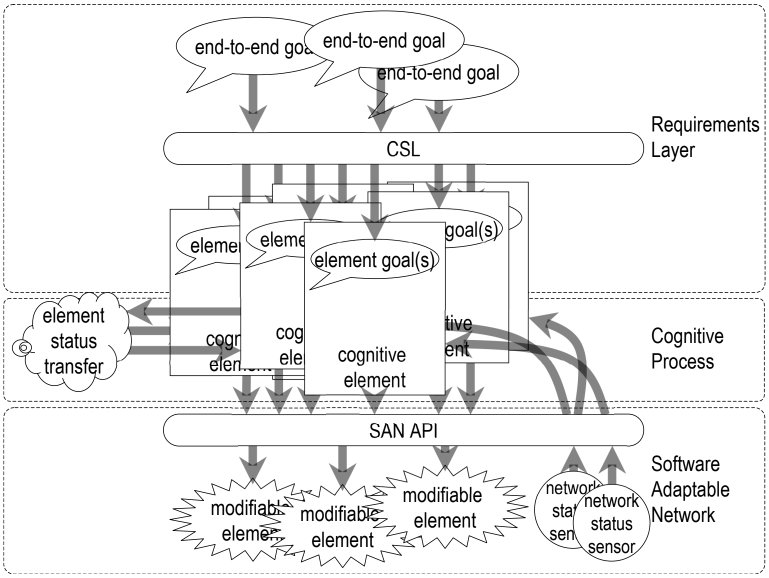


Figure 3.2: The three-layer cognitive framework, consisting of the requirements layer, cognitive process and Software Adaptable Network.

the radio. This difference in goal scope from end-to-end to local requires the CN to operate across all layers of the protocol stack.

Current research in CR emphasizes interactions with the PHY layer, which limits the impacts of changes made by the cognitive process to the radio itself and other radios in its range. Agreement with other radios on parameters which must match for successful link communication is reached through a process of negotiation. In CNs, the negotiations required extend beyond the nodes impacted by PHY and MAC layers to include nodes impacted by changes in higher layer protocols. Since CR goals are scoped locally, achieving agreement on any higher layer goals is likely to both be a slow process and result in sub-optimal performance at the network level. In contrast, CNs include these higher layer goals when making adaptations at lower network layers. However, the broader scope (as compared to CRs) of CNs may make reaching its goals more difficult.

The scope of CNs also reaches beyond the scope of cross-layer designs. The CN must support trade-offs between multiple goals, whereas cross-layer designs typically perform single objective optimizations. Cross-layer designs perform independent optimizations that may not account for the end-to-end performance goals. Trying to achieve each goal independently is likely to be sub-optimal, and as the number of cross-layer designs within a node grows, conflicts between the independent adaptations may lead to adaptation loops [46]. This pitfall is avoided in a CN by jointly considering all goals in the optimization process.

To connect the goals of the top-level users of the network to the cognitive process, an interface layer must be developed. In a CN, this role is performed by the CSL, providing behavioral guidance to the cognitive elements by translating the end-end goals to local element goals. This process is similar in some manners to the RKRL proposed by Mitola for CR, but differs in its core purpose. RKRL, as described in Chapter 2, derives from the KQML and is designed to provide a common knowledge language that allows radios to share information. CSL, on the other hand, is more analogous in scope and intention to QoS specification languages. There are already several different QoS specification paradigms in existence and the concept of these languages – mapping requirements to underlying mechanisms – is the same here, except that the mechanisms are adaptive to the network capabilities as opposed to a fixed set. In establishing criteria for a successful CSL, we adapt the criteria described by Jin et al. [29].

The following criteria are objectives for an effective CSL. It should be noted that these criteria are measures of a successful CSL, not requirements. A language that does not meet any or all of these requirements may still perform the role of a CSL, albeit less effectively.

- Expressiveness: A CSL must be able to specify a wide variety of end-to-end goals. It should be able to express constraints, goals, priorities and behaviors to the cognitive elements that make up the process. It should be able to express new goals without requiring a revision in the language.
- Cognitive process independence: The cognitive process architecture and functionality should not dictate the CSL. Instead, the CSL should abstract away as much of the cognitive process as possible to the application, user, or resource. This allows a goal

to be used with different cognitive processes with little modification and promotes re-usability.

- Interface independence: Whether the cognitive process is distributed or centralized in operation, autonomous or aggregated in architecture, the user should be presented as abstract an interface as possible. Like the previous criteria, this abstration promotes reusability by allowing the re-use of goals over many different cognitive processes with little effort from the top layer.

- Extensibility: The CSL should be extensible enough to adapt to new network elements, applications and goals, some of which may not even be imagined yet.

The scope of the entire cognitive process is broader than that of the constituent cognitive elements; it operates within the scope of a data flow, which may include many cognitive elements. These cognitive elements are the action elements of the cognitive process – they make the decisions that dictate the behavior of the network. As previously discussed, the cognitive process may consist of architectures as diverse as completely distributed and autonomous to centralized and monolithic. Because of the inherently complex and dynamic nature of the system it may not be feasible to find an exact optimal solution in highly dynamic network environments. Furthermore, the job of converting end-to-end goals to element goals is itself a difficult problem.

3.2.2 Cognitive Process

There does not seem to be a common, accepted definition of what cognition means when applied to communication technologies. The term cognitive, as used by this paper, follows closely in the footsteps of the definition used by Mitola [1] and the even broader definition of the FCC [35]. The former incorporates a spectrum of cognitive behaviors, from goal-based decisions to proactive adaptation. This definition of cognition is broader than the definitions of cognition used by most cognitive scientists, but it allows the inclusion of more limited and simple forms of cognition, such as adaptive systems.

Here, we associate cognition with *machine learning*, which is broadly defined by Thathachar [104] as any algorithm that “improves its performance through experience gained over a period of time without complete information about the environment in which it operates.” Underneath this definition, many different kinds of artificial intelligence, decision making and adaptive algorithms can be placed, giving CNs a wide scope of possible mechanisms to use in learning.

One hallmark of a cognitive process is the presence of learning and reasoning. We consider reasoning as described in [55]: the immediate decision process, using historical and current knowledge, that chooses the set of actions for the network. Learning, on the other hand, is the long-term process of accumulating knowledge from the results of past actions in order to improve the efficacy of future reasoning.

The actual reasoning process may take on many different forms, such as pure or hybrid versions of the problem solving models presented in Section 2.3.5, or some other model. Many of the differences in cognitive process architectures can be characterized by the same differences found in shared and distributed memory models of parallel processing architectures. For instance, learning automata that are distributed and autonomous can be analogous to a distributed memory model. Others, such as pMOOEAs, are distributed and parallelized monolithic processes that can behave in manners similar to a shared memory model. Like parallel processor architectures, however, there is a significant blurring between these areas, and many cognitive processes may act in manners somewhere in-between. In contrast to these distributed architectures, a single element cognitive process may be implemented as a centralized, single process.

Regardless of what reasoning method is chosen, the process needs to be able to converge to a solution faster than the network state changes. It must then re-learn and converge to new solutions when the status changes again. The issue of convergence is of particular importance in environments that change frequently, such as mobile wireless networks. Staying abreast of a dynamic environment is a difficult problem for many problem-solving techniques to handle. Discerning dynamic aspects of the network is difficult in large, complex environments. Furthermore, once the process has converged to a solution, many techniques need to be told that the system has changed in order to find a new solution. Some approaches, such as learning automata, can be modified so that they never completely converge to an operating state [105]. Instead, they remain statistically unstable, allowing the system to adapt to new changes as they occur.

The difficulty in adapting to dynamic network behavior makes architectures that use autonomous elements operating as rational agents appealing. A classic example of a system consisting of autonomous rational agents that meets socially-conscious goals is a capitalistic western-style economy. A capitalistic economy is made up of large numbers of autonomous entities operating in their best interest. With a small (relative to to the size of the economy) amount of control (in the form of legislation, money policy, and incentives), these individual, selfish actions can grow the economy to the benefit of all members in the economy. In contrast, countries such as North Korea, which attempt to directly control every aspect of the economy, often have economies that struggle. Trying to manage the entire complex economy directly is a difficult task.[1]

Rational, autonomous behaviors can lead to self-organization and emergent properties, which, although are not "order for free," may have a cost lower than attempting to create the order directly. Using mechanism design to create a "game" encouraging autonomous behavior that achieves or approaches the end-to-end goals is a problem of varying difficulty. Some objectives are naturally complementary to autonomous rational behavior. For instance, the end-to-end objective of maximizing the lifetime of a wireless network is complementary to the local radio goal of minimizing transmission power. Some objectives, are not as comple-

[1] It might be noted here that that even managing a western economy is an enormously difficult task, with the field of economics constantly revising its theories and models. However, the last fifty years have provided excellent empirical evidence that it is easier for a society to guide Smith's "invisible hand" then to attempt to take direct control of it.

mentary. For instance, the end-to-end objective of reducing the average flow delay is not complementary to a node's rational desire to prioritize its traffic above all others. Some sort of incentive structure, as discussed in Section 2.3.2 may be required to convince a node to transmit flows it did not originate with higher priority to reduce network delay.

Even rational agents with simple behaviors need the ability to learn from prior actions in order to make better future decisions. Learning serves to complement the objective optimization part of the cognitive process by retaining the effectiveness of past decisions under a given set of conditions. Determining the effectiveness of past decisions requires a feedback loop to measure the success of the chosen solution in meeting the objectives defined. This is retained in memory so that when similar circumstances are encountered in the future, the cognitive process will know where to start or what decisions to avoid.

The CN learns from prior decisions and applies the learning to future decisions. This is in contrast to most cross-layer designs, which are typically memoryless adaptations that will respond the same way when presented with the same set of inputs, regardless of how poorly the adaptation may have performed in the past. The ability to learn from past behavior is particularly important in light of the fact that the understanding of the interaction between layers is limited.

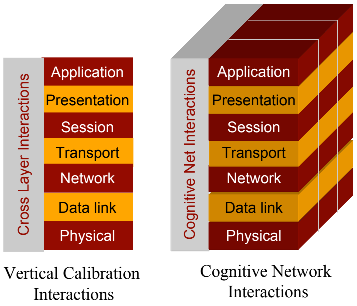
Cognitive element decisions can be implemented either synchronously or asynchronously. The details of making synchronous decisions across a large number of distributed elements with high reliability are likely to be complex. The implications of nodes switching configuration at different times may be worse than if no adaptation had been performed at all. Also, the varying topology of the network means that not all nodes will receive notification of configuration changes at the same time. One possible approach is to require elements to be synchronized to some common time reference and have the SAN issue configuration changes with respect to the time reference. This assumes that the SAN has an error free mechanism for communicating synchronization messages. Another approach is to have the individual elements make adaptations quickly compared to the time between adaptations, reducing the probability of asynchronous operations. Unfortunately, this approach can delay network actions, resulting lagging adaptations.

3.2.3 Software Adaptable Network

The SAN consists of the Application Programming Interface (API), modifiable network elements, and network status sensors. The SAN is really a separate research area, just as the design of the SDR is separate from the development of the CR but, at a minimum, the cognitive process needs to be aware of the interface and the network elements it controls. Just like the other aspects of the framework, the API should be flexible and extensible. Continuing the analogy with SDRs, an existing system that is analogous to the API is the Software Communications Architecture (SCA) used in the JTRS.

Unlike cross-layer design, which may span two or more layers in the network stack, CN APIs can span across nodes, tying together aspects of the network stack that may not normally be in the same stack. For instance, PHY layer parameters may be shared among several nodes not within physical range of one another, even if higher layer protocols (i.e. routing, transport) are not directly part of the adaptation. This gives end-to-end scope to all levels of the network stack, not just those that are naturally end-to-end in nature. A visual representation of the difference between CN and cross-layer interactions is shown in Figure 3.3.

Another responsibility of the API is to notify the cognitive process of the status of the network. The status of the network is the source of the feedback used by the cognitive process. These observations from the networks status sensors are distinct from any observations that might be shared internally to the cognitive process, which are a function of the cognitive process architecture. The level and detail of these observations depends on the filtering and abstraction being applied by the cognitive process. Possible observations may be local, such as bit error rate, battery life and data rate, or non-local, such as end-to-end delay and clique size. In the cognitive process, a cognitive element may compile these observations



Vertical Calibration Interactions

Cognitive Network Interactions

Figure 3.3: A visual representation of the differences between vertical calibration and CN interactions. CN interactions can include elements not directly in the network stack.

with observations from other elements to form a better model of non-local behavior. The availability and usage of this inter-element information is discussed at the end of Section 3.2.2.

The modifiable elements can include any parameter in a network, although it is unlikely that all parameters in a SAN would be modifiable. Each element should have public and private interfaces to the API, allowing it to be manipulated by both the SAN and the cognitive process. Modifiable elements are assumed to have a set of states that they can operate in. A solution for a cognitive process consists of a set of these states that, when taken together, meet the end-to-end requirements of the system. At any given instant the set of all possible combinations of states A can be partitioned into two subsets. The first, $A^{'}$, contains all possible combinations of decisions $\mathbf{a}$ that meet the end-to-end requirements and the second, $\bar{A}^{'}$, consists of all combinations that do not meet these requirements. Returning to the optimization discussion of Section 2.3.5, $A^{'}$ represents points in the feasible region of F. Those in $A^{'}$ that meet the requirements efficiently are a set of PO points, called A^{*}.

At the PHY and MAC layers, there may be conflict over what modifiable network elements a CR and a cognitive process control. One approach is to turn all control over to the cognitive process, but this is probably not wise. The reason is that the cognitive process has to limit its observations as much as possible just to make decisions about a complex network feasible. This leaves much detailed local information out of the cognitive process picture.

This detailed local information may be used by the CR to further optimize its performance outside the bounds of what is controlled by the CN. In order to allow this, the CR must know what it is allowed to change and what is in the hands of the CN. A potential solution is to allow the CN to establish regulatory policy for the CR in a real-time manner, leaving the CR to perform further optimization under the constraints established by the CN policy.

The SAN provides the foundation on which the rest of the CN gains its functionality and feedback. The flexibility, adaptability and functionality of the cognitive process is limited by the kind and amount of control that the SAN contains.

3.3 Critical Design Decisions

In order to identify the critical design decisions for a cognitive network, we return to the top-down, architecture-agnostic approach. The selfishness (defined as whether network elements pursue purely local objectives) of the decision-making elements in the cognitive network affect the system's *behavior*. The degree of ignorance these decision-making elements have with respect to current network conditions affects the quality of element decision *computations*. Finally, the amount of *control* that the decision-making elements have over the network affects the ability of the network to arrive at particular solutions. This section uses the game theoretic notation first described in Section 2.3.2.

3.3.1 Behavior

For cognitive processes with more than one cognitive element, the behaviors of the elements fall into a spectrum of behaviors ranging from purely selfish and individualistic to socially-conscious and altruistic. While altruistic behaviors may seem a natural method of accomplishing end-to-end goals, selfish behaviors sometimes lead to globally efficient adaptations. Furthermore, if the cognitive elements are autonomous and not under central control, selfish behaviors are a reasonable method of controlling the network, since real-world systems often consist of unrelated nodes with varying degrees of selfish behaviors. Additionally, as selfishness can require less coordination than altruism, it may lead to lower communication and processing overhead.

An action is selfish when the utility for element i, as a result of this action, given that every other element plays the actions that element i believes them to, is no less than the current utility for element i.

Definition 3.3.1 (Selfish Action)**.** *A selfish action is one in which:*

$$u_i(a_i^{t+1}, \tilde{a}_{-i}^{t+1}) \geq u_i(a_i^t, \tilde{a}_{-i}^{t+1})$$

where $\tilde{a}_i^{t+1}$ represents the action vector that element i believes the other elements will play.

Selfishness can be broken into two sub-categories: self-interested and egoistic cooperation (this term was first use in [106]). Self-interested selfishness operates to the detriment of the end-to-end objectives, whereas egoistic cooperation operates to the benefit of the local and end-to-end objectives.

The *belief* part of definition 3.3.1 is important, since ignorance may make this vector different than the actual strategies the other elements are employing. The network exhibits the feature of *selfishness* when every element plays a selfish strategy, meaning that all cognitive elements only select actions that continuously improve their individual objectives.

Altruistic strategies arise from the end-to-end objectives that the cognitive network is trying to achieve. An objective function, called *cost* and denoted by the function $C : A \to \mathbb{R}$, quantifies the performance of a network in achieving these goals for a specific action vector. This function is determined from the objectives of the network and, unlike the utility function, is improved as it is decreased.

An action is altruistic when an element sacrifices its utility to decrease the cost function.

Definition 3.3.2 (Altruistic Action). *An altruistic action is one in which:*

$$u_i(a_i^{t+1}, \tilde{a}_{-i}^{t+1}) < u_i(a_i^t, \tilde{a}_{-i}^{t+1})$$
$$C(a_i^{t+1}, \tilde{a}_{-i}^{t+1}) \leq C(a_i^t, \tilde{a}_{-i}^{t+1})$$

A network behavior exhibits the feature of *altruism* if at least one element exhibits an altruistic action at some point in the sequence of decisions.

3.3.2 Computation

The cognitive process requires some level of information from the system to perform its computations. Beyond direct observations, the outcome of previous decisions can also be used to learn what decisions are effective and to infer additional properties of the system. Ignorance in a cognitive network can come from one of two places: incomplete information and imperfect information. Incomplete information means that a cognitive element does not know the goals of the other elements or how much the other elements value their goals.

Imperfect information, on the other hand, means that the information about the other users' actions is unknown in some respect. We define Y_i as the set of signals that cognitive element i observes to learn about the actions of all elements in the network (including itself). The probability that action $\mathbf{a}$ was the action given that $y_i \in Y_i$ was observed is denoted by $P[\mathbf{a}|y_i]$.

In this work, we will assume all elements are of the same, known type. Thus the feature of *ignorance* occurs when at any stage in the decision sequence an ignorant decision is made.

Definition 3.3.3 (Ignorant Decision). *An ignorant decision occurs when a cognitive element has partial knowledge. This means for some i when there exists a $y_i \in Y_i$ such that $P[\mathbf{a}|y_i] < 1$ for all $\mathbf{a} \in A$.*

Ignorance occurs when at least one cognitive element in the decision sequence makes an ignorant decision. This can occur because of any of the following conditions:

- *Uncertain information*: The signal, as measured, has some random, stochastic uncertainty in it.
- *Missing information*: The signal is missing the action of at least one other element.
- *Indistinguishable information*: The signal may indicate one of several actions for the network, and it is impossible to distinguish between them.

If none of these conditions are present, then the network operates under fully observable actions, and has the feature of knowledge.

Definition 3.3.4 (Knowledgeable Decision)**.** *A knowledgeable decision occurs for some i when there exists an $\mathbf{a} \in A$ such that $P[\mathbf{a}|y_i] = 1$ for all $y_i \in Y_i$.*

Full knowledge occurs when all elements make knowledgeable decisions at every stage in the decision sequence.

3.3.3 Control

If there is cognitive element control over every action in the network, then the cognitive process could enact any particular solution to reach the end-to-end goals. Even if the cognitive process has less than full control over the actions, in some cases it may still be able to arrive at a set of desired system actions, depending on the cognitive process of the network.

With cognitive control over every multi-state element, the cognitive process can potentially set the system to any state. An ideal cognitive process could set the state to a state in A^*. If the system has only a few points of cognitive control or chooses not to exercise all its control, the cognitive process has to use the functionality and interactions of the non-cognitive aspects of the network to set the system state. Like the hole at the bottom of a funnel, certain system states will be basins of attraction, pulling the system towards them from a variety of starting states. If a system has several attractors and some are better than others, a few points of cognitive control may be enough to draw the system out of one attractor and into another. This is analogous to a watershed, in which moving the source of water a few miles may be enough to change what river the water will finally flow into.

If a cognitive element has control over one network parameter, under full control that parameter is under cognitive control across the entire network. For instance, full control over the radio transmission power means the cognitive elements have power control at every radio. Reducing cognitive control changes this symmetry. The feature of *partial-control* occurs when:

Definition 3.3.5 (Partial-control)**.** *For a cognitive network with k instances of the functionality in the network, and x instances of the functionality under cognitive control, partial-control occurs when*

$$\frac{x}{k} < 1$$

Definition 3.3.6 (Full-control)**.** *For a cognitive network with k instances of the functionality in the network, and x instances of the functionality under cognitive control, full-control occurs when*

$$\frac{x}{k} = 1$$

3.3.4 Price of a Feature

The above discussion has identified the features that each critical design decision can take on. A cognitive network with feature c is denoted $\mathcal{C}_c$. The set of action vectors $A_c \subseteq A$ represent the subset of possible action combinations that are fixed-points or basins of attraction for the cognitive network with that particular feature. Similarly, we define the set $A_{c'}$ to be the set of fixed-point and basin of attraction action vectors for the cognitive network $\mathcal{C}_{c'}$ with feature c' substituted for c, but all other aspects and features of the cognitive network remaining the same. Action vectors $\mathbf{a}_c$ and $\mathbf{a}_{c'}$ represent particular instances of the set of actions A_c and $A_{c'}$.

From an initial action vector $\mathbf{a}^0$ and following a sequence of decision making π, a cognitive network will transition to either a fixed point vector $\mathbf{a}_c^\pi \in A_c$ or a set of vectors making up a basin of attraction $A_c^\pi \subseteq A_c$. In the latter case we define $\mathbf{a}_c^\pi$ as the maximum-cost action vector in the basin of attraction, $\mathbf{a}_c^\pi = \arg\max_{\mathbf{a}\in A_c^\pi} C(\mathbf{a})$.

We define the metric *price of feature* c to be proportional to the ratio of the cost achieved by the network with c to that of the network with c'. Thus the price ρ can be written for the max-cost end-point action vector of each feature, considering initial action $\mathbf{a}^0$.

Definition 3.3.7 (Price of a Feature)**.** *The price of a feature ρ in a cognitive network with feature c compared to that with feature c' following order of play contained in π and initial action vector $\mathbf{a}^0$ is*

$$\rho(\mathbf{a}^0, \pi, \mathcal{C}_c, \mathcal{C}_{c'}) = \frac{C(\mathbf{a}_c^\pi) - C(\mathbf{a}_{c'}^\pi)}{C(\mathbf{a}_{c'}^\pi)}$$

Values of ρ greater than 0 indicate that the cognitive network has *worse* performance with feature c than with feature c'; values less than 0 indicate the cognitive network performs *better* with the new feature.

Definition 3.3.8 (Expected Price of a Feature)**.** *The expected price of a feature $\bar{\rho}$ in a cognitive network with feature c compared to that with feature c' is calculated over the set of all orders of play Π and all initial action vectors*

$$\bar{\rho}(A, \Pi, \mathcal{C}_c, \mathcal{C}_{c'}) = \underset{\mathbf{a}^0\in A, \pi\in\Pi}{\text{average}}\, \rho(\mathbf{a}^0, \pi, \mathcal{C}_c, \mathcal{C}_{c'})$$

Definition 3.3.9 (Bounded Price of a Feature)**.** *The bounded price of a feature P in a cognitive network with feature c compared to that with feature c' is calculated by determining the smallest upper bound over all possible initial actions and orders of decision making Π:*

$$P(A, \Pi, \mathcal{C}_c, \mathcal{C}_{c'}) = \sup_{\mathbf{a}^0\in A; \pi\in\Pi} \rho(\mathbf{a}^0, \pi, \mathcal{C}_c, \mathcal{C}_{c'})$$

The definition of the bounded price of a feature is consistent with the more problem-specific price of anarchy studied by Roughgarden and Tardos [107], although these authors use the ratio, rather than the relative change, of performance. Values of P represent the most possible deterioration in performance or the least possible improvement in performance for a feature tradeoff.

Clearly, more than just these three design decisions may affect the cost of a network. Furthermore, in reality it may be difficult to modify a cognitive network so that it only changes the feature within a particular design decision. There may be interactions between features, where changing one design decision may affect another. This can make it difficult to determine the price of just one feature. However, this metric, when measurable, quantifies the individual effects of each design decision.

3.4 Conclusion

This chapter has identified the CN problem by first defining and illustrating it before presenting a layered framework for implementation.The three layers of the framework - requirements, cognitive process, and SAN - mimic the three layer model for describing cognition. Each layer of the framework is left open in terms of design and implementation, with the description here acting as a functional specification. The features and functions of these layers will be used to guide the investigation of CNs in the rest of this research.

Chapter 4

Classes of Cognitive Networks

Certain cognitive networks are structurally similar to one another. The network and cognitive element objectives align in a manner that, when particular strategies are used by the cognitive elements, the network can be assured of converging to desirable network and cognitive element performance. When it is possible to identify this similar underlying structure, we say that the cognitive networks that share this similarity are part of the same class.

We examine classes of cognitive networks that require selfish strategies. The first class we identify is the potential class, which assures the convergence of the network to NE that are local-optima for network objectives. The second class we identify is the quasi-concave class, which assures the convergence of the network to a Pareto Optimal Nash Equilibrium (PONE) that is both a network and cognitive element optima. For each class, we present examples of real-world applications and identify the objectives of cognitive elements that are sufficiently aligned with the network objectives to ensure selfish strategies achieve the desired results.

4.1 Potential Class

A potential game is a normal form game that has a potential function that expresses the change in utility that every unilaterally deviating player experiences. Potential games are of interest since the utility behaviors of all players can be mapped into one function which means that the NE are found at some of the potential function's local optima.

Applications of potential games have been studied fairly intensively since their identification by Monderer and Shapley [108]. In particular, Neel examines the impacts of potential games in networks of cognitive radios in [10]. However, his research focused on the fact that these games reach a NE, with less attention given to the properties of this equilibrium with respect to the players and the network objectives.

Cognitive networks in the potential class include all cognitive networks that can be described as a normal form game with the potential game property. Since this is a very large set of networks, our examination focuses on those members with special NE properties, such as being PO or a local network optima.

All potential games require a potential function that expresses, with some amount of generality, the change in utilities observed by the players for a unilateral deviation in play. We begin by defining two commonly identified classes of potential games: Exact Potential Games (EPGs) and Ordinal Potential Games (OPGs). Exact Potential Functions (EPFs) express this change in utility exactly, while Ordinal Potential Functions (OPFs) express the change with the same sign, meaning positive changes in the potential function represents a positive change in utility, and vice-versa.

Definition 4.1.1 (Exact Potential Game)**.** *A normal form game $\Gamma = \langle N, A, u \rangle$ is an EPG if there exists an EPF $V : A \rightarrow \mathbb{R}$ such that*

$$V\left(a_i, a_{-i}\right) - V\left(b_i, a_{-i}\right) = u_i\left(a_i, a_{-i}\right) - u_i\left(b_i, a_{-i}\right)$$

for all $i \in N$, $a_{-i} \in A_{-i}$ and a_i, $b_i \in A_i$.

Definition 4.1.2 (Ordinal Potential Game)**.** *A normal form game $\Gamma = \langle N, A, u\rangle$ is an OPG if there exists an OPF $V : A \to \mathbb{R}$ such that*

$$V\left(a_i, a_{-i}\right) - V\left(b_i, a_{-i}\right) > 0 \Leftrightarrow u_i\left(a_i, a_{-i}\right) - u_i\left(b_i, a_{-i}\right) > 0$$

for all $i \in N$, $a_{-i} \in A_{-i}$ and a_i, $b_i \in A_i$.

EPGs are a subset of OPGs, meaning that every EPG is an OPG. Other identified classes of potential games include Weighted Potential Games (WPGs), Generalized ϵ-Potential Games (GϵPGs) and Generalized Ordinal Potential Games (GOPGs) where EPG $\subseteq$ WPG $\subseteq$ OPG $\subseteq$ GOPG. In this work, we concentrate particularly on OPGs.

Potential games are of interest because they contain several useful properties. All potential games are guaranteed to contain at least one NE [108], and the following lemma establishes how it can be identified:

Lemma 4.1.1. *Let Γ be a cognitive network that can be modeled as a finite normal form game and has an OPF $V : A \to \mathbb{R}$. The pure action vector $\mathbf{a} \in A$ is a NE if and only if for every $i \in N$*

$$V(\mathbf{a}) \geq V(a_i, a_{-i}) \forall a_{-i} \in A_{-i}$$

We are particularly interested in selfish strategies. Under all selfish strategies, potential games have the special property that they arrive at a NE. The general property identified in selfish strategies for potential games is the Finite Improvement Path (FIP):

Definition 4.1.3 (Finite Improvement Path)**.** *A FIP is a strategy of finite length in which the action vectors in the sequence $\{a^0, a^1, ldots\}$ deviates unilaterally at each step such that $u_i(a_i^{t+1}, a_{-i}^t) > u_i(a_i^t, a_{-i}^t)$.*

From this definition and Definition 3.3.1, we can see that when game play follows an improvement path, every player utilizes a selfish strategy.

For a cogntiive network in the potential class, when the potential function reflects the network-wide objective we say it has a cost-aligned potential function.

Definition 4.1.4 (Cost Aligned Potential Function)**.** *A cost aligned potential function preserves the ordering of the cost function.* $V : A \to \mathbb{R}$ *is a cost aligned potential function if there exists the following relationship*

$$V(\mathbf{a}) \geq V(\mathbf{b}) \Leftrightarrow C(\mathbf{a}) \leq C(\mathbf{b})$$

for every $\mathbf{a}, \mathbf{b} \in A$

This implies that if the cost function is aligned, then $-C(\mathbf{a})$ is an OPF for the game.

In the following section, we will identify an application of the potential class that have network objectives that are aligned with the potential function.

4.2 Applications of the Potential Class

The following two applications of the potential class are drawn from the wireless networking world. These applications consider an ad-hoc network consisting of a set of N wireless radios. Signal reception is modeled as a disc – depending on the transmission power, either a radio can connect to another radio, or it cannot. Antennas have an omnidirectional gain pattern, meaning that all radios within a radius proportional to $\sqrt[\alpha]{pt_i}$ of the source (where α is the path loss exponent and pt_i is the transmission power) of the radio are within reception range.

Radio transmission powers induce a topology that can be modeled as a directed connectivity multi-graph $G = (N, E)$, where directed arcs e_{ij} in E represent connections from a transmitter to a receiver. A bi-directional connection exists between two radios when arcs exist in both directions between the two. G is *connected* if there exists a bi-directed path—a collection of contiguous bi-directed arcs—between any two radios in G.

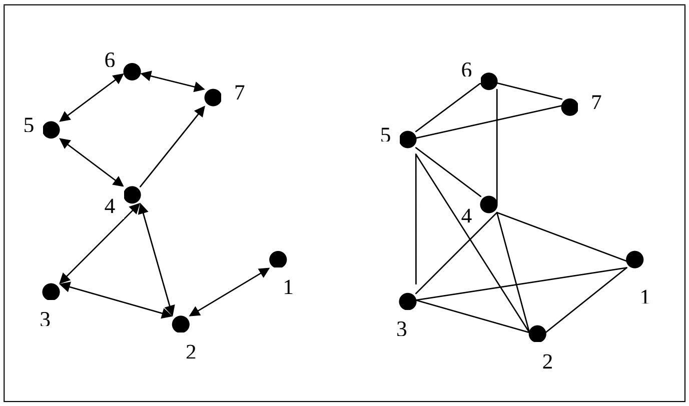


Figure 4.1: From a connectivity graph, G, (left), the conflict graph, U, is derived (right).

A topology G is full-duplex when the radios in the network are assigned to non-conflicting channels using Frequency Division Multiple Access (FDMA)[1]. Under FDMA, every radio can receive on all channels simultaneously if necessary; each radio can only transmit on a single channel, although it can use any of the channels. To model possible conflicts, G is mapped to an undirected conflict graph U. This conflict graph is created by placing an undirected edge between each pair of possible conflicting radios.

G is transformed into a modified distance-2 conflict graph U by placing undirected edges between all conflicting one-hop and two-hop neighbors, where two-hop neighbors share a common intermediary radio. If $U = (N, E')$, this transformation is given by $\mathfrak{T} : E \rightarrow E'$ where,

$$\mathfrak{T}(E) = \{e'_{ij} \mid e_{ij}, e_{ji} \in E \text{ or } e_{ik}, e_{ki}, e_{jk} \in E, \text{ for some } k \in N\} \tag{4.1}$$

For a visual illustration of this transformation, see Figure 4.1.

4.2.1 TopoPowerControl

Network connectivity is a fundamental requirement in topology control. For mobile wireless devices, the amount of transmission power required to become a part of the topology affects the lifetime of the device. The network objective is to maintain connectivity while minimiz-

[1] This analysis also applies to Time Division Multiple Access (TDMA), see [109] for a justification.

ing some characteristic of the network transmission powers, such as the sum power or the maximum power.

The TopoPowerControl problem is part of the potential class when it contains individual cognitive elements (the player set N) that selfishly adapt their transmission power (the action profile $\mathbf{pt}$) by choosing from the action set $A_i^{\text{TPC}} = (0, pt_i^{max}]$. The utility function reflects the cognitive element goal to maintain connectivity at the minimum transmission power:

$$u_i^{\text{TPC}}(\mathbf{pt}) = M f_i(\mathbf{pt}) - pt_i \tag{4.2}$$

Here, the function $f_i : \mathbb{R}^n \to \mathbb{Z}_+$ is the number of the radios that can be reached (possibly over multiple hops) by radio i via bidirectional connections and paths. The scalar benefit multiplier M_i indicates the value each radio places on being connected to other radios; we assume $M_i \geq \max\{pt_i^{max} \mid i \in N\}$.

Theorem 4.2.1. *The game* $\Gamma^{\text{TPC}} = \langle N, A^{\text{TPC}}, u^{\text{TPC}} \rangle$ *where the individual utilities are given by* (4.2) *is an OPG. An OPF is given by*

$$V^{\text{TPC}}(\mathbf{pt}) = M_i \sum_{i \in N} f_i(\mathbf{pt}) - \sum_{i \in N} pt_i \tag{4.3}$$

The proof is straightforward and was first given by Komali in [110]. The essence of the proof lies in the fact that $f_i(\mathbf{pt})$ is monotonic in $\mathbf{pt}$.

Proof. We prove by applying the definition of OPG from 4.1.2. First we have the following change in utility when changing from power pt_i to pt_i':

$$\begin{aligned} \Delta u_i &= u_i(pt_i, pt_{-i}) - u_i(pt_i', pt_{-i}) \\ &= M_i [f_i(pt_i, pt_{-i}) - f_i(pt_i', pt_{-i})] - (pt_i - pt_i') \end{aligned}$$

Similarly,

$$\begin{aligned}\Delta V &= V\left(pt_i, pt_{-i}\right) - V\left(pt'_i, pt_{-i}\right)\\ &= M_i\left[f_i\left(pt_i, pt_{-i}\right) - f_i\left(pt'_i, pt_{-i}\right)\right] - \left(pt_i - pt'_i\right)\\ &\quad +M_i\left[\sum\nolimits_{j\in N;j\neq i} f_i\left(pt_i, pt_{-i}\right) - f_i\left(pt'_i, pt_{-i}\right)\right]\end{aligned}$$

Therefore,

$$\Delta V = \Delta u_i + M_i\left[\sum_{j\neq i\in N} f_i\left(pt_i, pt_{-i}\right) - f_i\left(pt'_i, pt_{-i}\right)\right]$$

Since $f_i(\mathbf{pt})$ is monotonic and integer valued and $M_i \geq pt_i^{max}\ \forall i$, it follows that

$$\Delta u_i \begin{cases} \geq 0 \text{ if } pt_i > pt'_i \text{ and } f_i(\mathbf{pt}) > f_i\left(pt'_i, pt_{-i}\right);\\ \leq 0 \text{ if } pt_i < pt'_i \text{ and } f_i(\mathbf{pt}) < f_i\left(pt'_i, pt_{-i}\right);\\ < 0 \text{ if } pt_i > pt'_i \text{ and } f_i(\mathbf{pt}) = f_i\left(pt'_i, pt_{-i}\right);\\ > 0 \text{ if } pt_i < pt'_i \text{ and } f_i(\mathbf{pt}) = f_i\left(pt'_i, pt_{-i}\right)\end{cases}$$

Thus, the sign of the ΔV is the same as the sign of Δu_i, meaning that V is an OPF and Γ an OPG. □

If the end-to-end network objective is to minimize the total amount of power while keeping the network connected, then a cost function that characterizes this is:

$$C\left(\mathbf{pt}\right) = -M_i\sum_{i\in N} f_i\left(\mathbf{pt}\right) + \sum_{i\in N} pt_i \tag{4.4}$$

This function is higher for all partially connected topologies than for all connected topologies, and is minimized when the total transmission power is minimized for a connected topology. From Definition 4.1.4, it is trivial to see that since $V^{\text{TPC}}(\mathbf{pt}) = -C(\mathbf{pt})$ we have a cost aligned potential function. Thus selfish strategies are guaranteed to converge to local-optima for the network objective. However, it turns out (for this particular problem) that finding the *globally* minimum sum power that keeps the network connected is an NP-hard problem [111].

4.2.2 TopoChannelControl

Conventional MAC protocols (such as 802.11) perform poorly in multihop networks due to their inefficient spatial reuse. This motivates the channel allocation problem to avoid primary and secondary collisions, and thereby to improve spatial reuse.

The TopoChannelControl problem is part of the potential class when it consists of individual radios with FDMA capabilities making up the player set N, and each radio $i \in N$ picks a channel ch_i from a set of possible channels $A^{\text{TCC}} = \{0, 1, \ldots, \zeta\}$. The individual radios' utilities are given by:

$$u_i^{\text{CC}}(\mathbf{ch}) = \begin{cases} 1 & \text{iff } ch_i \notin \{ch_j \mid e_{ij} \in U\} \\ 0 & \text{otherwise.} \end{cases} \tag{4.5}$$

Theorem 4.2.2. *The game* $\Gamma^{\text{TCC}} = \langle N, A^{\text{TCC}}, u^{\text{TCC}} \rangle$ *where the individual utilities are given by* (4.5) *is an OPG. The OPF is given by*

$$V^{\text{TCC}}(\mathbf{ch}) = \sum_{i \in N} u_i(\mathbf{ch})$$

Proof. We need to show that $\forall i \in N$ and $\forall ch_i'$:

$$V(ch_i, ch_{-i}) - V(ch_i', ch_{-i}) > 0 \Leftrightarrow u_i(ch_i, ch_{-i}) - u_i(ch_i', ch_{-i}) > 0$$

Note that the change in potential function can be rewritten as:

$$\Delta V(\mathbf{ch}) = \Delta u_i(\mathbf{ch}) + \sum_{j \in \mathcal{N}_i^c; j \neq i} \Delta u_j(\mathbf{ch})$$

The utilities of the radios outside the neighborhood of i remain unaffected when radio i selects a new color.

Now, when i changes its action from ch_i to ch'_i (and thereby changes its utility from u_i to u'_i), three possibilities arise:

- $u_i = 1$ and $u'_i = 0$: In this case, i changes its color to one that matches at least one of its neighbor's color $ch_j \in ch_{\mathcal{N}_i^c}$. Therefore, the utility of j decreases (if it had utility of 1 to begin with) or remains the same (if it had a utility of 0). It follows that, $\Delta u_i(\mathbf{ch}) < 0$ implies $\sum_{j\in\mathcal{N}_i^c; j\neq i} \Delta u_j(\mathbf{ch}) \leq 0$. Hence, $\Delta u_i(\mathbf{ch}) < 0$ implies $\Delta V(\mathbf{ch}) < 0$.
- $\Delta u_i = 0$: In this case, the conflicting status of the color chosen by i doesn't change. Therefore, the utility of every radio j remains unaffected. It follows that, $\Delta u_i(\mathbf{ch}) = 0$ implies $\sum_{j\in\mathcal{N}_i^c; j\neq i} \Delta u_j(\mathbf{ch}) = 0$. Hence, $\Delta u_i(\mathbf{ch}) = 0$ implies $\Delta V(\mathbf{ch}) = 0$.
- $u_i = 0$ and $u'_i = 1$: In this case, i originally has the same color as that of at least one of its neighbors. When i selects a new color distinct from that of all its neighbors, it obtains a utility $u'_i = 1$. Therefore, the utility of every radio $j \in \mathcal{N}_i^c$ would increase (if j's color matched only with that of i to begin with) or would remain same (if j's did not match i's original selection). In either case, $\Delta u_j(\mathbf{ch}) \geq 0$. Therefore, $\Delta u_i(\mathbf{ch}) > 0$ implies $\Delta V(\mathbf{ch}) > 0$.

For all three cases, $\operatorname{sgn}(\Delta V) = \operatorname{sgn}(\Delta u_i)$. Therefore, V is an OPF and the game is an OPG. □

If the network objective is to have conflict-free channel assignment, the cost function can be represented by

$$C(\mathbf{ch}) = -\sum_{i\in N} u_i(\mathbf{ch}) \tag{4.6}$$

It is trivial to see the potential function is cost-aligned.

If we assume that the number of channels ζ is equal to the number of radios ($|N|$), then we can show that at steady-state the network accomplishes this network objective.

Lemma 4.2.3. *If there are sufficient number of channels available, every NE state is a conflict-free channel assignment.*

Proof. Proof by contradiction is immediate. If a radio conflicts with another radio, it can unilaterally improve its utility by choosing a color different from its neighbors (this is possible because there are enough colors ζ in the palette). This contradicts the supposition that the state is a NE. □

Furthermore, under this assumption, this steady state is PO.

Lemma 4.2.4. *If there are sufficient number of channels available, every NE state is PO.*

Proof. From Lemma 4.2.2, we see that no radio can further improve its utility, since all radios are conflict free. Thus the state is PO. □

4.3 Quasi-Concave Class

We now focus on a class of cognitive networks that is more restrictive in its classification than the potential class. Specifically, we identify types of cognitive networks that have a single, unique NE that is also PO, and, like the potential class, converges to this NE action vector under selfish strategies.

We call this class the quasi-concave class because of the shape of the utility function in the action space. There is some overlap between the potential class and the quasi-concave class, but the quasi-concave class guarantees the *uniqueness* and *Pareto optimality* of the NE, something that the more general potential class cannot.

We say a cognitive network belongs to the quasi-concave class when it can be modeled as a normal form game $\Gamma = \langle N, A, u_i \rangle$ where:

1. A is convex subset of $\mathbb{R}$;

2. u_i is continuous in A and quasi-concave in a_i;

3. u_i^{max} is constant with respect to $\mathbf{a}\, \forall \mathbf{a} \in A$ where $u_i^{max} = \max_{a_i \in A_i} u_i(a_i, a_{-i})$.

The first property uses the term quasi-concavity, which can be defined as:

Definition 4.3.1 (Quasi-Concavity)**.** *A function is quasi-concave when there exists an* $a'_i \in A_i$ *such that for all* $a_i < a'_i \in A_i$*,* $u_i(a_i, a_{-i})$ *is non-decreasing and for all* $a_i > a'_i \in A_i$*,* $u_i(a_i, a_{-i})$ *is non-increasing.*

The second property states that the maximum value for every u_i is constant, regardless of the actions of the other cognitive elements. Graphically, this implies that if u_i is the "height" of a particular action vector for a cognitive element, then u_i produces a multi-dimensional "ridge" of constant value across all actions.

Cognitive networks of this class call for selfish strategies. The best-response selfish action is found from the best-response function $\mathbf{z} : A \to A$ which is given by:

$$\mathbf{z}(\mathbf{a}) = \arg\max_{a_i \in A_i} u_i(a_i, a_{-i}) \; \forall i \in N \tag{4.7}$$

Whereas the potential class required strategies that have the FIP property, the quasi-concave class does not make this requirement. This means the selfish strategies may note finite but rather asymptotically convergent. Furthermore, the better response strategy requires that the action space be some convex subset of $\mathbb{R}$. Under this requirement, we can describe the strategy of better-response actions formally as:

$$a_i^{t+1} = (1 - \alpha^t)a_i^t + \alpha^t z_i(\mathbf{a}^t) \tag{4.8}$$

where $0 < \alpha^t \leq 1$, $t = \{0, 1, 2, \ldots\}$.

This says that each element will play some convex combination of its best response action and the last action. Since this class of cognitive networks have quasi-concave utility functions, any selection of α^t will lead to a better response action, meaning the better-response strategy described in (4.8) includes several more specific strategies, such as best response ($\alpha^t \equiv 1$), random better response, exhaustive better response, and intelligent better response. The latter three strategies are called *decision rules* and their operation is described in more detail in [10].

Lemma 4.3.1. *If Γ has a pure strategy NE, the NE action vector $\hat{\mathbf{a}}$ results in $u_i(\hat{\mathbf{a}}) = u_i^{max}\ \forall i \in N$.*

Proof. Assume the NE action vector has an element with a utility less than u_i^{max}. Thus for a NE action vector $\mathbf{a}'$ there exists i in N such that $u_i' < u_i^{max}$. Recall u_i is continuous in A and quasi-concave in a_i with u_i^{max} constant with respect to $a_{-i}\ \forall a_{-i} \in A_{-i}$. Thus if $u_i' < u_i^{max}$, regardless of a_{-i}, there is always an a_i that improves u_i. For the action vector $\mathbf{a}'$, there is an improving action for the element to choose, and thus $\mathbf{a}'$ is not a NE. We have found a contradiction, meaning that the NE action vector $\mathbf{a}$ occurs at $u_i^{max}\ \forall i \in N$. □

Theorem 4.3.2. *If Γ has a pure strategy NE, this action vector is a PONE.*

Proof. This is from inspection given Lemma 4.3.1: the NE action vector $\hat{\mathbf{a}}$ occurs at $u_j^{max} \forall j \in N$. Since u_i^{max} is constant with respect to $a_j \forall j \neq i \in N$, there cannot exist any action vector $\mathbf{a}'$ with a utility greater than $\hat{\mathbf{a}}$ for some $j \in N$. Thus $\hat{\mathbf{a}}$ is both a NE and PO. □

Figure 4.3 shows an example of a two element system that meets these properties.

Selfish behavior can achieve end-to-end goals in two cases: when the selfish behavior is aligned with the end-to-end goal or when incentive structures are used to force an alignment between the two. For the potential class, we identified an example of when the potential function was cost-aligned. Now we identify utility functions that are cost aligned.

Definition 4.3.2 (Strict Dominance)**.** *Vector $\mathbf{a}$ strictly dominates vector $\mathbf{b}$ if $a_i > b_i\ \forall i$. This is represented by $\mathbf{a} \succ \mathbf{b}$.*

Definition 4.3.3 (Weak Dominance)**.** *Vector $\mathbf{a}$ weakly dominates vector $\mathbf{b}$ if either $a_i \geq b_i$ $\forall i$ or there exists exists at least one $a_i > b_i$ and one $a_j < b_j$. This is represented by $\mathbf{a} \succeq \mathbf{b}$.*

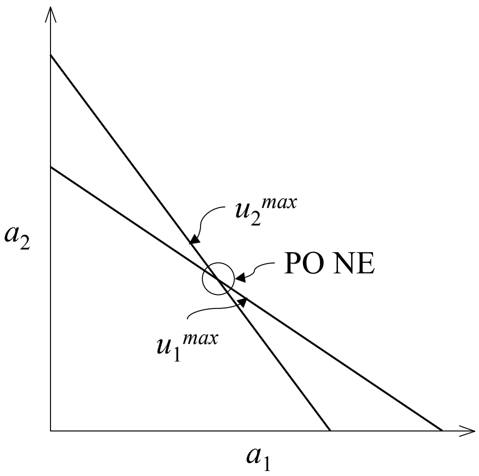


Figure 4.2: A graphical representation of the quasi-concave class. The lines represent u_i^{max}, the peak of u_i and thus the "best response" action of a_i for any $\mathbf{a}$.

Definition 4.3.4 (Cost Aligned Utility Function Space). *A utility space $u : A \to \mathbb{R}^n$ is cost-aligned if, for every $\mathbf{a}, \mathbf{b} \in A$:*

$$\textit{if } u(\mathbf{a}) \prec u(\mathbf{b}) \textit{ then } C(\mathbf{b}) < C(\mathbf{a})$$

Here the relational operator $\prec$ means that the right side strictly dominates the left; in our case, if $u \prec u'$, then $u_i < u'_i$ for every i. The relational operator $\preceq$ means that the left side is dominated or unordered in comparison to the right side. Unordered means that for some u and u', there exists at least one i such that $u_i < u'_i$ and at least one j such that $u_j > u'_j$.

Example utility functions spaces that are cost aligned include the following, as well as their linear transformations:

$$\begin{aligned} C(\mathbf{a}) &= -\textstyle\sum_{i \in N} u_i(\mathbf{a}) \\ C(\mathbf{a}) &= -\min_{i \in N} u_i(\mathbf{a}) \\ C(\mathbf{a}) &= -\max_{i \in N} u_i(\mathbf{a}) \end{aligned} \tag{4.9}$$

We identify three sets of convergence requirements for the quasi-concave class from the literature: Uryasev, Gabay and Yates. Each set of convergence requirements places different restrictions on the strategies, utilities and action spaces of the cognitive elements to insure asymptotic convergence to a PONE.

Uryasev's Convergence We begin with a general form for an asymptotically stable cognitive network, from Uryasev and Rubinstein [112]. Following their method, we use a global function, the Nikaido-Isoda function [113], to turn the cognitive network into a maximization problem.

We begin by defining the Nikaido-Isoda function $\Psi : A \times A \rightarrow \mathbb{R}$:

$$\Psi(\mathbf{a}, \mathbf{b}) = \sum_{i \in N} [u_i(b_i, a_{-i}) - u_i(\mathbf{a})] \tag{4.10}$$

This means that the Nikaido-Isoda function is the summation of the change in utility for each player when they unilaterally change their action from a_i to b_i. From this definition, we observe the following property of the NE. If $\hat{a}$ is a NE, then:

$$\max_{\mathbf{a} \in A} \Psi(\hat{\mathbf{a}}, \mathbf{a}) = 0 \tag{4.11}$$

The best response function can also be re-written using the Nikaido-Isoda function as follows:

$$\mathbf{z}(\mathbf{a}) = \arg\max_{\mathbf{b} \in A} \Psi(\mathbf{a}, \mathbf{b}) \tag{4.12}$$

Before describing the properties that the Uryasev convergence requires, we define a few terms:

Definition 4.3.5 (Weakly Convex [114])**.** *Let $f : X \times X \rightarrow \mathbb{R}$ be a function defined on a Cartesian product $X \times X$ where X is a convex closed subset of the Euclidean space $\mathbb{R}^n$. Further, we consider that $f(x, z)$ is weakly convex on X with respect to the first argument, meaning*

$$\alpha f(\mathbf{x}, \mathbf{z}) + (1 - \alpha) f(\mathbf{y}, \mathbf{z}) \geq f(\alpha \mathbf{x} + (1 - \alpha)\mathbf{y}, \mathbf{z}) + \alpha(1 - \alpha) r_z(\mathbf{x}, \mathbf{y})$$

for all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in X$*;* $\alpha \in (0, 1]$ *where the remainder function* r_z *has the property:*

$$\frac{r_z(\mathbf{x}, \mathbf{y})}{\|\mathbf{x} - \mathbf{y}\|} \to 0, \text{ as } \|\mathbf{x} - \mathbf{y}\| \to 0$$

for all $\mathbf{x} \in X$*.*

Definition 4.3.6 (Weakly Concave [114])**.** *Let* $f : X \times X \to \mathbb{R}$ *be a function defined on a Cartesian product* $X \times X$ *where* X *is a convex closed subset of the Euclidean space* $\mathbb{R}^n$*. Further, we consider that* $f(z, y)$ *is weakly concave on* X *with respect to the second argument, meaning*

$$\alpha f(\mathbf{z}, \mathbf{x}) + (1 - \alpha) f(\mathbf{z}, \mathbf{y}) \leq f(\mathbf{z}, \alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) + \alpha(1 - \alpha)\mu_z(\mathbf{x}, \mathbf{y})$$

for all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in X$*;* $\alpha \in (0, 1]$ *where the remainder function* μ_z *has the property:*

$$\frac{\mu_z(\mathbf{x}, \mathbf{y})}{\|\mathbf{x} - \mathbf{y}\|} \to 0, \text{ as } \|\mathbf{x} - \mathbf{y}\| \to 0$$

for all $\mathbf{z} \in X$*.*

If a function $f(\mathbf{x}, \mathbf{y})$ satisfies both Definition 4.3.5 and 4.3.6, it is called *weakly convex-concave.*

Uryasev convergence requires the following additional properties beyond Γ :

1. A is a convex, compact subset of $\mathbb{R}^n$;

2. the Nikaido-Isoda function is continuously weakly convex-concave;

3. the residual term $r_{\mathbf{c}}(\mathbf{a}, \mathbf{b})$ of Definition 4.3.5 for $\Psi(\mathbf{a}, \mathbf{b})$ is uniformly continuous on A with respect to $\mathbf{c}$ for all $\mathbf{a}, \mathbf{b} \in A$;

4. the residual terms satisfy

$$r_{\mathbf{b}}(\mathbf{a}, \mathbf{b}) - \mu_{\mathbf{a}}(\mathbf{a}, \mathbf{b}) \geq \beta(\|\mathbf{a} - \mathbf{b}\|) \; \forall \mathbf{a}, \mathbf{b} \in A \tag{4.13}$$

where $\beta(0) = 0$ and β is a strictly increasing function;

5. the best response function $\mathbf{z}(\mathbf{a})$ is single-valued and continuous on A; and
6. the better-response function meets the following requirements:
   (a) $\sum_{t=0}^{\infty} \alpha^t = \infty$,
   (b) $\alpha^t \to 0$ as $t \to \infty$, and
   (c) actions are chosen synchronously and universally.

Quasi-concave cognitive networks that meet these properties will be denoted Γ^{Ψ}.

These properties mean various things about the cognitive network. The first property indicates the action space is a continuous, bounded, subset of the reals. This means that the modifiable aspects of the cognitive network contain continuously adjustable parameters such as energy, power, time, size, or cost, as opposed to discretely adjustable parameters such as channel, state or route. The second through fifth properties describe the manner that the utility functions of the cognitive elements interact. The final property describes the better-response strategy. This property places specific restrictions on the better response strategy as described in 4.8, eliminating the best-response strategy ($\alpha \equiv 1$) and requires that all cognitive elements synchronously update their actions using the same universal α^t parameter.

These properties are more common than may appear at first glance. The family of weakly convex-concave functions includes the family of smooth functions (those for which derivatives of all finite orders exist). Thus property 2 can be satisfied if $\Psi(\mathbf{a}, \mathbf{b})$ is smooth. The continuousness of $\Psi(\mathbf{a}, \mathbf{b})$ ensures property 3. If $\Psi(\mathbf{a}, \mathbf{b})$ is twice-differentiable (which it is if it is smooth), then property 4 can be satisfied by function $\mathbf{Q} : A \times A \to \mathbb{M}(n, n)$:

$$\mathbf{Q}(\mathbf{a}, \mathbf{a}) = \Psi_{\mathbf{aa}}(\mathbf{a}, \mathbf{b})|_{\mathbf{b}=\mathbf{a}} - \Psi_{\mathbf{bb}}(\mathbf{a}, \mathbf{b})|_{\mathbf{b}=\mathbf{a}} \tag{4.14}$$

being positive definite, where $\Psi_{\mathbf{aa}}(\mathbf{a}, \mathbf{b})|_{\mathbf{b}=\mathbf{a}}$ is the Hessian of the Nikaido-Isoda function with respect to the first argument and $\Psi_{\mathbf{bb}}(\mathbf{a},\mathbf{b})|_{\mathbf{b}=\mathbf{a}}$ is the Hessian of the Nikaido-Isoda function with respect to the second argument, both evaluated at $\mathbf{b}=\mathbf{a}$.

Theorem 4.3.3. *The better-response strategy on* Γ^{Ψ} *converges to a unique PONE, regardless of the initial action vector.*

This result comes from combining the result of Theorem 4.3.2 with the significant results of [112]. The sketch of the proof from [112] is as follows: since $\Psi(\mathbf{a},\mathbf{b})$ is a weakly convex-concave function and the residual terms satisfy property 4, improving action vectors always move the system towards the maximum of the Nikaido-Isoda function in a contracting manner. Property 4 also implies the existence and uniqueness of the NE.

Gabay's Convergence Gabay [115] introduces a simpler method of determining asymptotic stability, but adds restrictions by assuming that $A=\mathbb{R}^n_+$ (meaning that a_i is in the interval $[0,+\infty)$ for all $i\in N$).

We define two matrices, $\mathbf{f}$ and $\mathbf{f}'$, where $\mathbf{f}$ is the partial differential matrix

$$f_i(\mathbf{a})=\frac{\partial u_i}{\partial a_i}(\mathbf{a}),\ i=1,2\ldots n \tag{4.15}$$

and $\mathbf{f}'$ is the Jacobian matrix of u where

$$f'_{ij}(\mathbf{a})=\frac{\partial^2 u_i}{\partial a_i\partial a_j}(\mathbf{a}),\ i,j=1,2\ldots n \tag{4.16}$$

Definition 4.3.7 (Coercivity)**.** *A function is coercive if*

$$\lim_{a_i\to+\infty}|f(\mathbf{a})|=+\infty$$

$\forall i\in N.$

Definition 4.3.8 (Strictly Diagonally Dominant)**.** *A matrix* $\mathbf{m}$ *is called strictly diagonally* *dominant if:*

$$|m_{ii}| > \sum_{j \neq i \in N} |m_{ij}|$$

$\forall i \in N$.

The quasi-concave networks denoted Γ^+ have the following additional properties beyond those of Γ:

1. $A = \mathbb{R}^n_+$;
2. $f_i(\mathbf{a})$ is coercive $\forall i \in N$;
3. $f'_i(\mathbf{a})$ exists $\forall i \in N$; and
4. $\mathbf{f}'$ is strictly diagonally dominant for all $\mathbf{a} \in A$.

Unlike quasi-concave networks with the properties of Γ^Ψ, Γ^+ networks has actions that are in the set of all the positive reals. This is both a more restrictive and less restrictive property: $\mathbb{R}^+$ is an open set (something that Γ^Ψ does not allow), but it only allows one specific open set. Most continuously modifiable aspects of real systems are closed, so finding cognitive networks that fit Γ^+ exactly will be difficult. On the other hand, the last three properties are much simpler to verify than those in Γ^Ψ.

Theorem 4.3.4. *The better-response strategy on Γ^+ converges to a unique PONE, regardless of the initial action vector.*

This result comes from combining the result of Theorem 4.3.2 with the significant results of Gabay in [115]. A sketch of the proof from [115] can be made as follows: since $\mathbf{f}'$ is strictly diagonally dominant, $|f'_{ii}(\mathbf{a})| > 0$ and u_i has a unique maximizer in a_i (recall u_i is continuous in a and quasi-concave in a_i). Furthermore (since $A = \mathbb{R}^n_+$) every $a_i^t \in \mathbb{R}_+$, when put into the better-response process, leads to a $a_i^{t+1} \in \mathbb{R}_+$. The strict diagonal dominance shows that this mapping of a_i^t to a_i^{t+1} is contracting, and when combined with the coercivity property, leads to the convergence of the system to a single NE[2].

Yate's Convergence Yates[3] uses a set of convergence properties related to the best response function $\mathbf{z}(\mathbf{a})$.

The following properties extend those of Γ for networks with Yates convergence, denoted Γ^z.

1. The best response is positive, $\mathbf{z}(\mathbf{a}) > 0 \, \forall \mathbf{a} \in A$;
2. the best response is monotonic, $\mathbf{a} \succ \mathbf{b}$ then $\mathbf{z}(\mathbf{a}) \succ \mathbf{z}(\mathbf{b})$;
3. the best response is scalable, meaning that for all $\phi > 1$, $\phi\mathbf{z}(\mathbf{a}) \succ \mathbf{z}(\phi\mathbf{a})$;
4. the NE point $\hat{\mathbf{a}}$ exists in A; and
5. the better-response process has $\alpha^t = 1$ for all t.

Theorem 4.3.5. *The better-response strategy on converges to a unique PONE, regardless of the initial action vector.*

This result comes from combining the result of Theorem 4.3.2 with the significant results of Yates in [116]. A sketch of the proof from [116] can be made as follows: The monotonicity and scalability properties imply that the NE $\hat{\mathbf{a}}$ is unique. If we select $\tilde{\mathbf{a}}^0$ such that $\tilde{\mathbf{a}}^0 \succ \mathbf{z}(\tilde{\mathbf{a}}^0)$, it is apparent that $\tilde{\mathbf{a}}^t$ is a monotonic decreasing series converging to $\hat{\mathbf{a}}$. If we select $\bar{\mathbf{a}}^0$ such that $\bar{\mathbf{a}}^0 = \mathbf{0}$, it is apparent that $\bar{\mathbf{a}}^t$ is a montonic increasing series converging to $\hat{\mathbf{a}}$. Choosing an arbitrary $\mathbf{a}^0$ places this action vector somewhere between these two converging series, meaning that it too will converge to $\hat{\mathbf{a}}$.

The similarities and differences between these three convergence requirements are summarized in Table 4.1.

[2] It is interesting to note the relationship between the better-response strategy and Gabay's convergence and the Jacobi and Gauss-Seidel methods of solving systems of equations. In this case, the better-response strategy is solving the system equations defined by $\mathbf{f}(\mathbf{a}) = 0$.

[3] Technically, Yates doesn't use a game theoretic model. Neel makes perhaps the first mapping of Yate's ideas to a game theoretical framework in [10].

	$A \subseteq \mathbb{R}^n$	A is compact convex	$A = \mathbb{R}^n_+$	$u(\mathbf{a})$-focused	$\mathbf{z}(\mathbf{a})$-focused	all better-response strategies
Uryasev's convergence	●	●		●		
Gabay's convergence	●		●	●		●
Yate's convergence	●				●	

Table 4.1: A summary of features for each of the convergence requirements.

4.4 Applications of the Quasi-Concave Class

We now use the convergence results from the previous section to derive sufficiency conditions for two classes of cognitive network problems.

4.4.1 Average Chasing

Several quasi-concave applications can be placed under the broader taxonomy of being "average chasing." An element in an average chasing network attempts to align its weighted action to a weighted sum of all other cognitive element actions. These networks have cognitive element utilities that are in the following form:

$$u_i(\mathbf{a}) = -\left(\left(c_i + \sum_{j \neq i \in N} w_{ji} a_j\right) - w_{ii} a_i\right)^2 \tag{4.17}$$

where A_i is a non-empty convex subset of $\mathbb{R}$. By inspection, u_i is continuous in A and quasi-concave in a_i, u_i^{max} is constant with respect to a_j $\forall j \neq i \in N$.

Uryasev's Convergence

Proposition 4.4.1. *If A is a compact convex subset of $\mathbb{R}^n$, the better response function meets the requirements of Item 6 and $|2w_{ii}^2| > \sum_{j\neq i\in N} |w_{ij}w_{ii} + w_{jj}w_{ji}|$ $\forall i \in N$, then the average chasing category converges under Uryasev.*

Proof. We prove by going through each item of Uryasev's convergence:

1. From assumption.

2. The Nikaido-Isoda function is given by:

$$\begin{aligned} \Psi(\mathbf{a},\mathbf{b}) \;=\; & \textstyle\sum_{k\in N}\Big\{-\Big(\Big(c_k + \sum_{l\neq k\in N} w_{lk}a_l\Big) - w_{kk}b_k\Big)^2 \\ & + \Big(\Big(c_k + \sum_{l\neq k\in N} w_{lk}a_l\Big) - w_{kk}a_k\Big)^2\Big\} \end{aligned} \tag{4.18}$$

   $\Psi(\mathbf{a},\mathbf{b})$ is a polynomial with respect to a_i, b_i for all $i \in N$, making it part of the family of smooth functions and thus weakly convex-concave.

3. $\Psi(\mathbf{a},\mathbf{b})$ is continuous, ensuring the residual term continuousness.

4. $\Psi(\mathbf{a},\mathbf{b})$ is polynomial, thus twice differential. $\mathbf{Q}(\mathbf{a},\mathbf{a})$ is given by:

$$Q_{ij}(\mathbf{a},\mathbf{a}) = \begin{cases} 4w_{ii}^2 & i = j \\ -2w_{ij}w_{ii} - 2w_{jj}w_{ji} & \text{otherwise} \end{cases} \tag{4.19}$$

   Since $\mathbf{Q}$ is symmetric, if it is diagonally dominant it is positive definite (note that this property is sufficient, but not necessary). This occurs when:

$$\left|2w_{ii}^2\right| > \sum_{j\neq i\in N} |w_{ij}w_{ii} + w_{jj}w_{ji}| \;\forall i \in N \tag{4.20}$$

   which is from the assumption.

5. The best response function is:

$$\mathbf{z}(\mathbf{a}) = \frac{c_i + \sum_{j \neq i \in N} w_{ji} a_j}{w_{ii}} \; \forall i \in N \tag{4.21}$$

   which is single valued and continuous in A.

6. From the assumption[4].

□

Note that the only constants that affect the convergence of average-chasing under Γ^{Ψ} are the weights; the additive constants make no difference. Furthermore, instead of being independent, all the weight values are inter-dependent, with convergence dependent on how the identity and non-identity weights from each cognitive element utility relate.

Gabay's Convergence

Proposition 4.4.2. *If A is $\mathbb{R}^n_+$ and $|w_{ii}| > \sum_{j \neq i \in N} |w_{ji}|$ then the average chasing category converges under Gabay's.*

Proof. We prove by going through each item of Gabay's convergence

1. From assumption.

2. The coerciveness of $\partial u_i / \partial a_i$ is apparent from

$$\lim_{a_i \to +\infty} \left| \frac{\partial u_i}{\partial a_i}(\mathbf{a}) \right| = \lim_{a_i \to +\infty} \left| -2w_{ii} \left((c_i + \sum_{j \neq i \in N} w_{ji} a_j) - w_{ii} a_i \right) \right| = +\infty \tag{4.22}$$

[4] This sequence might not provide the most optimal method for reaching the PONE, see [114] for work on determining the optimal selection of α^t.

3. The matrix $\mathbf{f}(\mathbf{a})$ is:

$$f_i(\mathbf{a}) = \frac{\partial u_i}{\partial a_i}(\mathbf{a}) = -2w_{ii}\left((c_i + \sum_{j \neq i \in N} w_{ji}a_j) - w_{ii}a_i\right) \tag{4.23}$$

Thus the Jacobian, $\mathbf{f}'(\mathbf{a})$ is thus

$$f'_{ij} = \begin{cases} -2w_{ii}^2 & i = j \\ -2w_{ii}w_{ji} & \text{otherwise} \end{cases} \tag{4.24}$$

which exists for all $\mathbf{a}$ and i.

4. From Definition 4.3.8 we know that $\mathbf{f}'$ is diagonally dominant when:

$$|w_{ii}| > \sum_{j \neq i \in N} |w_{ji}| \tag{4.25}$$

which is from the assumptions.

□

This result tells us that the sum of the non-identity weights must be less than the identity weights to ensure convergence to the unique NE. Unlike Uryasev's convergence results, convergence is assured independently, with each utility function weight only dependent on the other weights in the function.

Yates' Convergence

Proposition 4.4.3. *If A is a compact, convex subset of $\mathbb{R}^n_+$, $w_{ij} > 0$ for all i, j, $c_i > 0$ for all i, and $\alpha^t \equiv 1$ then the average chasing category converges under Yates'.*

Proof. We prove by going through each item of Yates' convergence

1. Positivity is assured since $c_i > 0$, $w_{ij} > 0$, and A is a compact, convex subset of $\mathbb{R}^n_+$.

2. Monotonicity is assured since if $\mathbf{a}' \succ \mathbf{a}$ then $\sum_{j \neq i \in N} w_{ij} a'_j > \sum_{j \neq i \in N} w_{ij} a_j$ for every $i \in N$ and $\mathbf{z}(\mathbf{a}') \succ \mathbf{z}(\mathbf{a})$.

3. The scalability assumption means that, for all $\phi > 1$ that $\phi \mathbf{a} \in A$,

$$\begin{array}{rccc} & \phi \mathbf{z}(\mathbf{a}) & \succ & \mathbf{z}(\phi \mathbf{a}) \\ \rightarrow & \frac{\phi c_i + \phi \sum_{j \neq i \in N} w_{ji} a_j}{w_{ii}} & > & \frac{c_i + \sum_{j \neq i \in N} w_{ji} \phi a_j}{w_{ii}} \ \forall i \in N \end{array}$$

   Since $c_i > 0$, inspection shows that scalability holds.

4. We can promise that a pure strategy NE $\hat{\mathbf{a}}$ exits in A. Since A is a compact, convex subset of $\mathbb{R}^n_+$ and u_i is quasi-concave with respect to a_i, from the Glicksberg-Fan fixed point theorem [117, 118], Γ^z has a fixed point and thus a pure strategy NE in A.

5. From assumption.

□

PowerControl

Perhaps the most-common real-world example of average chasing is the PowerControl problem. This problem has been discussed many times [116, 41, 119] in varying forms. Unlike TopoPowerControl, which attempts to maintain the topological connectivity, PowerControl attempts to maintain the link connectivity. Radios have designated receivers that they transmit to, attempting to reach a target SINR requirement at.

The network objective is to find the point at which all transmitting radios meet the required SINR at the receiving radios while simultaneously meeting some network transmission power objective, such as minimizing the maximum or total transmission power in the network. The set of transmitting radios is N, and each selects as their transmission power pt_i from the action set A_i^{PC}, which is some subset of $\mathbb{R}^+$. Assuming that some degraded connection is possible at SINRs lower than the target, a utility that expresses the cognitive elements' desire to transmit as closely as possible to the receiver's SINR requirement is:

$$u_i^{\mathrm{PC}}(\mathbf{pt}) = -\left(\left(\sum_{k\neq i\in N} g_{kj}pt_k + \sigma_j\right)\gamma_j - g_{ij}pt_i\right)^2 \tag{4.26}$$

where $\mathbf{g}$ is the matrix of gains between radio i and j, j is the intended receiver for radio i, γ_j is the goal SINR ratio at the receiver, and σ_j is the thermal noise at the receiver.

Using Uryasev's convergence, $\mathbf{Q}(\mathbf{pt},\mathbf{pt})$ must be positive definite to have property 4 of Γ^{Ψ}. To avoid confusion, instead of using i and j as the transmitter-receiver pair, we use the notation d_i to represent the intended receiver for some radio i.

$$Q_{ij}(\mathbf{pt},\mathbf{pt}) = \begin{cases} 4g_{id_i}^2 & i=j \\ -2g_{id_i}g_{jd_i}\gamma_{d_i} - 2g_{jd_i}g_{id_i}\gamma_{d_j} & \text{otherwise} \end{cases} \tag{4.27}$$

This is a symmetric constraint, meaning that positive definiteness is assured by strict diagonal dominance of $\mathbf{Q}(\mathbf{pt},\mathbf{pt})$

$$\left|2g_{id_i}^2\right| > \sum_{j\neq i\in N} \left|g_{id_i}g_{jd_i}\gamma_{d_i} + g_{jd_j}g_{id_j}\gamma_{d_j})\right| \ \forall i \in N \tag{4.28}$$

If we simplify the assumptions so that $pt \in \mathbb{R}_+^n$ to use Gabay's convergence, we get the constraint on $\mathbf{f}'$ that:

$$f'_{ij}(\mathbf{pt}) = \begin{cases} 2g_{id_i}^2 & i=j \\ -2g_{id_i}g_{jd_i}\gamma_{d_i} & \text{otherwise} \end{cases} \tag{4.29}$$

meaning that the following must hold to ensure convergence by theorem 4.3.4:

$$|g_{id_i}| > \sum_{j\neq i\in N} |g_{jd_i}\gamma_{d_i}| \ \forall i \in N \tag{4.30}$$

Yates uses this problem [116] as the case study for his convergence theorem. The properties that $g_{ij} > 0$ for all i and j is supported by the properties of the gain terms. Furthermore, the property that $\sigma_j > 0$ is supported by the fact that thermal noise is non-zero and positive. Finally transmission power must be positive and bounded, making the action space a compact convex subset of $\mathbb{R}^n_+$.

Now is a good time to make a note about the potential real-world limitations of this mathematical model. These convergence requirements assume the action space falls into some continuous convex subset of the reals, which translates to an infinitely granular transmission power setting. Unfortunately, no real world systems truly has this fine control, with systems having a discrete and finite number of transmission power states. Because of this discretization, the feasible region of the transmission power state may not have a satisfying action vector.

If the network objective is to minimize the maximum transmission power in the network, subject to meeting the SINR constraint at each receiver, the cost function can be represented by the maximum of all the utilities:

$$C(\mathbf{pt}) = \max_{i \in N} \left(\left(\sum_{k \neq i \in N} g_{kj} pt_k + \sigma_j \right) \gamma_j - g_{ij} pt_i \right)^2 \tag{4.31}$$

By inspection, the utility function space is cost aligned (see (4.9)). Thus the network objective of minimizing the maximum transmission power while maintaining the SINR property can be approached by selfishly following the cognitive element goals.

SizeControl

Another quasi-concave application that falls under average chasing is transport layer security. If the network goal is to obfuscate the applications being used in the network and the kind of transactions occurring in those applications, one way to accomplish this is to encrypt the packets and normalize the packet lengths (similar to the function that packet padding can perform for point-to-point connections in Secure Shell (SSH) [120]). The network objective is to find the minimum packet length that allows all sources to appear to have identical average packet lengths.

The cognitive elements at each application make up the set of players N. Each element has control over the expected length of the pad, $E(pl_i)$ which is drawn from the action space, A_i^{SC}, a subset of the positive reals $\mathbb{R}_+$[5]. Each radio attempts to pad packets such that the expected length of the whole packet is slightly larger than the average of the expected length of the packets from each of the rest of the nodes in the network. The utility of the each cognitive element can be expressed as:

$$u_i^{\mathrm{SC}}(E(\mathbf{pl})) = -\left(\sum_{j \neq i \in N} \left(\frac{1}{n-1} E(pl_j + bl_j) \right) - ((1+\epsilon)E(pl_i) + E(bl_i)) \right)^2 \tag{4.32}$$

where bl_i is the body length of a packet from element i. Re-arrangement of the utility yields:

$$u_i(E(\mathbf{pl})) = -\left(\left(\left(\sum_{j \neq i \in N} \frac{1}{n-1} E(bl_j) - E(bl_i) \right) + \sum_{j \neq i \in N} \frac{1}{n-1} E(pl_j) \right) - (1+\epsilon)E(pl_i) \right)^2$$

This is clearly the average chasing form, as the first term is the local constant c_i (we assume that the applications generate, at steady state, a packet stream with a constant average body length).

For this application, we assume the expected padding lengths can be arbitrarily long. It is easy to show convergence using Gabay's convergence. Gabay's convergence is assured by the fact that SizeControl is average chasing, having all the properties of Γ^+, including the diagonal dominance property 4.

$$|1+\epsilon| > \sum_{j \neq i \in N} \left| \frac{1}{n-1} \right| = 1 \tag{4.33}$$

Thus the application will converge to the minimum packet length that is $1+\epsilon$ times longer than the average of all other node's packet length.

[5] The pad length at any given instant is drawn from the set of positive integers $\mathbb{Z}^+$.

If the network objective is to minimize the total amount of traffic in the network subject to obfuscation, the cost function can be written as:

$$C(E(\mathbf{pl})) = \sum_{i \in N} \left(\sum_{j \neq i \in N} \left(\frac{1}{n-1} E(pl_j + bl_j) \right) - ((1+\epsilon)E(pl_i) + E(bl_i)) \right)^2 \tag{4.34}$$

Again, by inspection, the utility space is aligned with this cost function (see (4.9)).

4.4.2 Product Chasing

Another category of the quasi-concave class applications is product chasing. Product chasing attempts to choose an action whose weighted value is close to the weighted product of all other element's actions. A general form for the utility of the cognitive elements is:

$$u_i(\mathbf{a}) = -\left(\left(c_i - \prod_{j \neq i \in N} (d_{ij} - w_{ij} a_j) \right) - w_{ii} a_i \right)^2 \tag{4.35}$$

By inspection, u_i is continuous in A and quasi-concave in a_i, u_i^{max} is constant with respect to a_j $\forall j \neq i \in N$.

Uryasev's Convergence

Proposition 4.4.4. *If A is a compact, convex subset of $\mathbb{R}^n_+$, the better response function meets the requirements of Item 6 and $w_{ij} > 0$ for all i, j, $c_i > 0$ for all i, and $|2w_{ii}^2| > \sum_{j \neq i \in N} \left| w_{ii} \prod_{l \neq i, j \in N} (d_{il} - w_{il} a_l) + w_{jj} \prod_{l \neq i, j \in N} (d_{jl} - w_{jl} a_l) \right|$ $\forall i \in N$ then the product chasing category converges under Uryasev's.*

Proof. We prove by going through each item of Uryasev's convergence:

1. The Nikaido-Isoda function is given by:

$$\begin{aligned}\Psi(\mathbf{a},\mathbf{b}) &= \textstyle\sum_{k\in N}\left\{-\left(\left(c_k-\prod_{l\neq k\in N}\left(d_{lk}-w_{lk}a_l\right)\right)-w_{kk}b_k\right)^2\right.\\ &\left.+\left(\left(c_k-\prod_{l\neq k\in N}\left(d_{lk}-w_{lk}a_l\right)\right)-w_{kk}a_k\right)^2\right\}\end{aligned} \tag{4.36}$$

   $\Psi(\mathbf{a},\mathbf{b})$ is a polynomial with respect to a_i, b_i for all $i\in N$, making it part of the family of smooth functions and thus weakly convex-concave.

2. $\Psi(\mathbf{a},\mathbf{b})$ is continuous, ensuring the residual term continuousness.

3. $\Psi(\mathbf{a},\mathbf{b})$ is polynomial, thus twice differential. $\mathbf{Q}(\mathbf{a},\mathbf{a})$ is given by:

$$Q_{ij}(\mathbf{a},\mathbf{a})=\begin{cases} 4w_{ii}^2 & i=j\\ -2w_{ii}\prod_{l\neq i,j\in N}\left(d_{il}-w_{il}a_l\right)-2w_{jj}\prod_{l\neq i,j\in N}\left(d_{jl}-w_{jl}a_l\right) & \text{otherwise}\end{cases} \tag{4.37}$$

   Since $\mathbf{Q}$ is symmetric, if it is diagonally dominant it is positive definite (note that this property is sufficient, but not necessary). This occurs when:

$$\left|2w_{ii}^2\right| > \sum_{j\neq i\in N}\left|w_{ii}\prod_{l\neq i,j\in N}\left(d_{il}-w_{il}a_l\right)+w_{jj}\prod_{l\neq i,j\in N}\left(d_{jl}-w_{jl}a_l\right)\right| \ \forall i\in N \tag{4.38}$$

   which is from the assumption.

4. The best response function is:

$$\mathbf{z}(\mathbf{a})=\frac{c_i-\prod_{j\neq i\in N}\left(d_{ij}-w_{ij}a_j\right)}{w_{ii}}\ \forall i\in N \tag{4.39}$$

   which is single valued and continuous in A.

5. From the assumption.

□

Gabay's Convergence

Proposition 4.4.5. *The product chasing category does not converge under Gabay's.*

Proof. Item 4 does not hold. $\mathbf{f}'$ is given by:

$$f'_{ij}(\mathbf{pt}) = \begin{cases} -2w_{ii}^2 & i = j \\ -2w_{ii}w_{ij}\prod_{k\neq i,j\in N}(d_{ik} - w_{ik}a_k) & \text{otherwise} \end{cases} \tag{4.40}$$

Which, to be strictly diagonally dominant requires:

$$|w_{ii}| > \sum_{j\neq i\in N}\left|w_{ij}\prod_{k\neq i,j\in N}(d_{ik} - w_{ik}a_k)\right| \tag{4.41}$$

Which is not possible for all $\mathbf{a} \in \mathbb{R}^n_+$, so product changing does not converge under Gabay's.

□

Yates' Convergence

Proposition 4.4.6. *If d_{ij}, $w_{ij} > 0$ for all i, j, A is a compact, convex subset of $\mathbb{R}^n_+$ such that $a_i \in [0, a_i^{max}]$ where $a_i^{max} = \max_{j\neq i\in N} d_{ij}/w_{ij}$, $c_i > \max_{j\neq i\in N} d_{ij}$ for all i, and $\alpha^t \equiv 1$ then the product chasing category converges under Yates'.*

The proof here is inspired by Yang in [121].

Proof. We prove by going through each item of Yates' convergence:

1. Positivity is assured since $(d_{ij} - w_{ij}a_j) < 1$ and $c_i > \max_{j\neq i\in N} d_{ij}$, making $\mathbf{z}(\mathbf{a}) > 0$ for all $a \in A$.

2. Monotonicity is assured since if $\mathbf{a}' \succ \mathbf{a}$ then $\left(d_{ij} - w_{ij}a'_j\right) < (d_{ij} - w_{ij}a_j)$ and $\mathbf{z}(\mathbf{a}') \succ \mathbf{z}(\mathbf{a})$.

3. The scalability assumption means that, for all $\phi > 1$ that $\phi\mathbf{a} \in A$,

$$\begin{aligned} & \phi\mathbf{z}(\mathbf{a}) - \mathbf{z}(\phi\mathbf{a}) & \succ & \quad \mathbf{0} \\ \rightarrow \quad & \frac{\phi c_i - \phi\prod_{j\neq i\in N}(d_{ij}-w_{ij}a_j) - c_i + \prod_{j\neq i\in N}(d_{ij}-w_{ij}\phi a_j)}{w_{ii}} & > & \quad 0 \, \forall i \in N \end{aligned}$$

We can show by induction that scalability holds. Begin with $|N| = 1$ (for readability, we will omit the w_{ii} term from the denominator as it is just a positive constant).

$$\begin{aligned} & \phi c_i - \phi \prod\nolimits_{j=1}(d_{ij} - w_{ij}a_j) - c_i + \prod\nolimits_{j=1}(d_{ij} - w_{ij}\phi a_j) & & \\ = \; & c_i(\phi - 1) - \phi(d_{i1} - w_{i1}a_1) + (d_{i1} - w_{i1}\phi a_1) & & \\ = \; & (c_i - 1)(\phi - 1) & > & \; 0 \end{aligned}$$

Now assume that for $|N| = k$, $c_i(\phi - 1) - \phi\prod_{j=1}^{k}(d_{ij} - w_{ij}a_j) + \prod_{j=1}^{k}(d_{ij} - w_{ij}\phi a_j) > 0$ and show it is true for the $|N| = k + 1$ case.

$$\begin{aligned} & \phi c_i - \phi \prod\nolimits_{j=1}^{k+1}(d_{ij} - w_{ij}a_j) - c_i + \prod\nolimits_{j=1}^{k+1}(d_{ij} - w_{ij}\phi a_j) & & \\ = \; & \phi c_i - \phi \prod\nolimits_{j=1}^{k}(d_{ij} - w_{ij}a_j)(1 - w_{ik+1}a_{k+1}) - c_i & & \\ & + \prod\nolimits_{j=1}^{k}(d_{ij} - w_{ij}\phi a_j)(d_{ik+1} - w_{ik+1}\phi a_j) & & \\ = \; & c_i(\phi - 1) - \phi \prod\nolimits_{j=1}^{k}(d_{ij} - w_{ij}a_j) + \prod\nolimits_{j=1}^{k}(d_{ij} - w_{ij}\phi a_j) & & \\ & + (w_{ik+1}\phi a_{k+1})\left(\prod\nolimits_{j=1}^{k}(d_{ij} - w_{ij}a_j) - \prod\nolimits_{j=1}^{k}(d_{ij} - w_{ij}\phi a_j)\right) & > & \; 0 \end{aligned}$$

where the positivity of the last line is assured from the k assumption and the fact that $(d_{ij} - w_{ij}a_j) > (d_{ij} - w_{ij}\phi a_j)$.

4. We can promise that a pure strategy NE $\hat{\mathbf{a}}$ exits in A. Since A is a compact, convex subset of $\mathbb{R}_{+}^{n}$ and u_i is quasi-concave with respect to a_i, from the Glicksberg-Fan fixed point theorem [117, 118], Γ^z has a fixed point and thus a pure strategy NE in A.

5. From assumption.

□

WindowControl

At the MAC layer, this class of cognitive network can be used to determine the contention window size for IEEE 802.11. Yang examines and solves this problem in [121]. Her network attempts to schedule the transmission of packets in a manner that still meet the individual delay properties of each message type. The cognitive elements utilities in this network can be represented by

$$u_i(\mathbf{v}) = -\left(\frac{H_2}{\Delta_i - H_1}\left(1 + \gamma - \prod_{j \in n_i}(1 - G_{ji}v_j)\right) - v_i\right)^2 \tag{4.42}$$

Here, v_i is the contention window size for flow i. Δ_i is the delay property for that flow. The variables γ, H_1, H_2 and G_{ji} are constants unrelated to the choice of contention window size.

Using Uryasev's convergence, the constraints on the network to converge to a PONE are that $\mathbf{Q}(\mathbf{v}, \mathbf{v})$ be positive definite.

$$Q_{ij}(\mathbf{v}, \mathbf{v}) = \begin{cases} 4 & i = j \\ -2\frac{H_2 G_{ii}}{\Delta_i - H_1}\prod_{l \neq i, j \in N}(1 - G_{il}v_l) - 2\frac{H_2 G_{jj}}{\Delta_i - H_1}\prod_{l \neq i, j \in N}(1 - G_{jl}v_l) & \text{otherwise} \end{cases} \tag{4.43}$$

We can use diagonal dominance of this symmetric matrix to assure the strict positivity, meaning that

$$2 > \sum_{j \neq i \in N}\left|\frac{H_2 G_{ii}}{\Delta_i - H_1}\prod_{l \neq i, j \in N}(1 - G_{il}v_l) + \frac{H_2 G_{jj}}{\Delta_i - H_1}\prod_{l \neq i, j \in N}(1 - G_{jl}v_l)\right| \tag{4.44}$$

In her paper [121], Yang proves asymptotic stability using Yates' convergence. Her work to show that the network meets all assumptions is not trivial and we refer the reader to the paper.

Yang's utility takes on the product chasing form because it is stochastically based; the product represents the probability that several events (transmissions) *do not* occur simultaneously. Other quasi-concave cognitive networks that require the use of probability may also be product chasing.

4.5 Analysis

We now analyze the quasi-concave and potential classes from the perspective of the critical design decisions described in Chapter 3. The bounded price of a feature, defined in Definition 3.3.9 is over all orders of evaluation π for the cognitive network. This is not a problem for both Gabay and Yates' convergence, which reach the same NE under all orders of evaluation. Uryasev's convergence, on the other hand, in Property 6 specifies it must have synchronous action selection. In the following analysis, we will abuse the definition of the bounded price of a feature when examining Uryasev convergence in quasi-concave networks by determining the maximum price over all initial action vectors, but not all orders of evaluation.

4.5.1 Price of Selfishness

To determine the price of selfishness for these classes, we return to the strategies employed. Recall that the potential class has the FIP property and that the better-response strategy will always converge in the quasi-concave class. From definition 3.3.1, we observe that both strategies are selfish.

A short side note on the relationship between the set of better-response strategies to selfish strategies: while inspection shows all better-response strategies are selfish, regardless of the sequence of α^t, Uryasev convergence has specific properties for this sequence ($\sum_{t=0}^{\infty} \alpha^t = \infty$ and $\alpha^t \to 0$ as $t \to \infty$). This eliminates, among other strategies, the best-response selfish strategy ($\alpha^t \equiv 1$ for all t). In contrast, Yates' convergence limits the the set better-response strategies to only the best-response strategy. Only Gabay's convergence allows the full set of better-response strategies.

For the quasi-concave class, if the utility space is cost aligned, the NE action vector is not just PO and unique in u, it is also optimal for the cost function. This also means that the cost of a cooperative strategy can never be lower than that of the selfish strategy, since that would mean that the cooperative strategy would have found a strategy that dominates the PO selfish strategy – an impossibility. Thus since $C(\hat{\mathbf{a}}^{\pi}_{selfish}) \leq \max_{\mathbf{a} \in A^{\pi}_{cooperative}} C(\mathbf{a})$, regardless of $\mathbf{a}^0$, the bounded price of selfishness for both average chasing and product chasing classes is 0.

For the potential class, if the potential function is cost aligned, the price of selfishness is not bounded in a general sense. While the NE will be locally-optimal with respect to the network objective, there is no guarantee about how close to the global optimum these points are. For instance, in TopoPowerControl the cognitive elements arrive at a local minimum sum power vector, but determining the globally minimum sum power vector is an NP-hard problem. Thus the power vector arrived at by TopoPowerControl can be arbitrarily worse than what some cooperative technique could arrive at, and the price of selfishness for the potential class is unbounded. However, in Chapter 6 we will present a selfish strategy for TopoPowerControl that optimizes the network objective of minimizing the maximum transmission power in the network. Thus it is possible to find the bounded price of selfishness for specific problems.

4.5.2 Price of Ignorance

The price of ignorance depends on the public signal used by the cognitive elements. For average chasing and product chasing in the quasi-concave class, if Y_i is equal to the sum or product of the weighted actions of all other players the cognitive elements have ignorance from indistinguishable information presented.

Under these conditions, the observed signal is either

$$y_i = c_i + \sum_{i \neq j \in M} w_{ij} a_j \tag{4.45}$$

for the average chasing class or

$$y_i = c_i - \prod_{i \neq j \in M} (d_{ij} - w_{ij} a_j) \tag{4.46}$$

for the product chasing class. Equation 4.45 represents the signal that is received in PowerControl, since each element receives as a signal the aggregate interference and noise received, not the individual power selections of the other radios in the network. Similarly, (4.46) represents the ignorance experienced by WindowControl, where the elements observe the average contention delay in combination with current contention window sizes . In both cases, the signal operates under indistinguishable ignorance since the individual actions of other elements are aggregated into one signal.

With either signal, the utility function can be written as

$$u_i(\mathbf{a}) = -(y_i - w_{ii} a_i)^2 \tag{4.47}$$

This is functionally the same as the original utility functions, meaning that $\hat{\mathbf{a}}^{\pi}_{ignorant} = \hat{\mathbf{a}}^{\pi}_{knowledge}$ and thus $C(\hat{\mathbf{a}}^{\pi}_{ignorant}) = C(\mathbf{a}^{\pi}_{knowledge})$, regardless of $\mathbf{a}^0$, and the bounded price of ignorance is 0. Of course, other forms of ignorance may have different prices and may not be bounded.

4.5.3 Price of Control

The price of control assumes that some fraction of the cognitive elements utilize a fixed-action strategy. We define $M \subseteq N$ to be the set of cognitively controlled functionality, and $M' = \{M \backslash N\}$ to be the set of non-cognitively controlled functionality.

Average Chasing Under partial control, the utility function is changed from that shown in (4.23) to the following

$$u_i(\mathbf{a}) = -\left(\left(c_i + e_i + \sum_{j \neq i \in M} w_{ji} a_j +\right) - w_{ii} a_i\right)^2 \tag{4.48}$$

where $e_i = \sum_{j \in M'} w_{ji} a_j$, and is a constant since all variables are constant including the action choices for every j in M'. If the original properties of Γ have not changed from this modification, then according to all three convergence theorems, the system will still converge to a PONE.

Examining Urayesev's convergence, we see that the Nikaido-Isoda function is now given by:

$$\begin{aligned} \Psi(\mathbf{a}_M, \mathbf{b}_M) \;\; = \;\; & \textstyle\sum_{k \in M} \Bigg\{ -\Big(\Big(c_k + e_k + \sum_{l \neq k \in M} w_{lk} a_l\Big) - w_{kk} b_k\Big)^2 \\ & + \Big(\Big(c_k + e_k + \sum_{l \neq k \in M} w_{lk} a_l\Big) - w_{kk} a_k\Big)^2 \Bigg\} \end{aligned} \tag{4.49}$$

where $\mathbf{a}_M \in A^M$ and $A^M = \times_{i \in M} A_i$.

Thus the smoothness property still exists, and the best response function is now:

$$\mathbf{z}(\mathbf{a}_M) = \frac{c_i + e_i + \sum_{j \neq i \in N} w_{ji} a_j}{w_{ii}} \;\forall i \in M \tag{4.50}$$

which is still single valued and continuous in A^M.

At this point it should be apparent that $\mathbf{Q}(\mathbf{a}_M, \mathbf{a}_M)$ is a submatrix of $\mathbf{Q}(\mathbf{a}, \mathbf{a})$. In particular, when an element is not under cognitive control, both that row and column are removed from $\mathbf{Q}(\mathbf{a}, \mathbf{a})$, leaving $\mathbf{Q}(\mathbf{a}_M, \mathbf{a}_M)$ also symmetric. If $\mathbf{Q}(\mathbf{a}, \mathbf{a})$ was diagonally dominant, then $\mathbf{Q}(\mathbf{a}_M, \mathbf{a}_M)$ is also diagonally dominant, since

$$\left|2w_{ii}^2\right| > \sum_{j \neq i \in N} \left|w_{ij} w_{ii} + w_{jj} w_{ji}\right| \;\forall i \in N$$

then since $M \subseteq N$

$$\left|2w_{ii}^2\right| > \sum_{j \neq i \in M} |w_{ij}w_{ii} + w_{jj}w_{ji}| \ \forall i \in M$$

and the remaining cognitively controlled elements still converge to a unique PONE. If the cost function is monotonic with respect to the utility space, we can assign a hierarchy to the bounded price of control.

We know that full-control of the CN will lead to a unique PO point, or u_i^{max} for every $i \in N$, and if the utility space is cost aligned, this will lead to an optimal global. If the network is under no control, the worst case occurs when every element chooses the action that gives the completely dominated utility vector $\mathbf{u}$ with the worst-case network performance. As we add control from this worst-case cost, we see that we generate dominating utility vectors: since each controlled cognitive element will converge to u_i^{max} and each non-cognitive controlled element can be no greater than this, if $|M_1| < |M_2|$, then $u_i(\mathbf{a}_{M_1} \cup \mathbf{a}_{M_1'}) \leq u_i(\mathbf{a}_{M_2} \cup \mathbf{a}_{M_2'}) \forall i \in N$.

A similar argument can be made for Gabay's and Yates' convergence. For Gabay's convergence, if $f'(\mathbf{a})$ is diagonally dominant under full control, then, because $M \subseteq N$, $f'(\mathbf{a}_M)$ will also be diagonally dominant and we will obtain the same hierarchy of bounded price of partial control. Yates' assumptions of positivity, monotonicity and scalability of the best response function are also not invalidated by (4.50), and assumming a $\hat{\mathbf{a}}$ still exists in A (a requirement for all convergence cases, since this derived from the initial assumptions on Γ), the system will contain the hierarchy of bounded prices.

Product Chasing Under partial-control, the utility function for product chasing in (4.35) becomes:

$$u_i(\mathbf{a}_M) = -\left(\left(c_i + e_i \prod_{j \neq i \in M} (d_{ij} - w_{ij}a_j)\right) - w_{ii}a_i\right)^2 \tag{4.51}$$

where $e_i = \prod_{j \in M'} (d_{ij} - w_{ij}\hat{a}_j)$ and is constant.

$$\begin{aligned}\Psi(\mathbf{a}_M, \mathbf{b}_M) \;&=\; \textstyle\sum_{k \in N} \Big\{ -\Big(\Big(c_k + e_k \prod_{l \neq k \in M} (d_{lk} - w_{lk}a_l)\Big) - w_{kk}b_k\Big)^2 \\ &\quad + \Big(\Big(c_k + e_k \prod_{l \neq k \in N} (d_{lk} - w_{lk}a_l)\Big) - w_{kk}a_k\Big)^2 \Big\}\end{aligned} \tag{4.52}$$

$\Psi(\mathbf{a}, \mathbf{b})$ is a polynomial with respect to a_i, b_i for all $i \in N$, making it part of the family of smooth functions. The best response function is:

$$\mathbf{z}(\mathbf{a}) = \frac{c_i + e_i \prod_{j \neq i \in N} (d_{ij} - w_{ij}a_j)}{w_{ii}} \;\forall i \in N \tag{4.53}$$

which is single valued and continuous in A.

The critical condition is thus showing the positive-definiteness of $\mathbf{Q}(\mathbf{a}_N, \mathbf{a}_N)$ for all $\mathbf{a}_N \in A^N$. Unlike the average-chasing partial control problem, $\mathbf{Q}(\mathbf{a}_N, \mathbf{a}_N)$ is not a true submatrix of $\mathbf{Q}(\mathbf{a}, \mathbf{a})$. It is given by:

$$Q_{ij}(\mathbf{a}, \mathbf{a}) = \begin{cases} 4w_{ii}^2 & i = j \\ -2w_{ii}e_i \prod_{l \neq i, j \in N} (d_{il} - w_{il}a_l) - 2w_{jj}e_j \prod_{l \neq i, j \in N} (d_{jl} - w_{jl}a_l) & \text{otherwise} \end{cases} \tag{4.54}$$

which is still a symmetric matrix. Positive-definiteness and convergence is assured by diagonal dominance, meaning that

$$\left|2w_{ii}^2\right| > \sum_{j \neq i \in N} \left| e_i w_{ii} \prod_{l \neq i, j \in N} (d_{il} - w_{il}a_l) + e_j w_{jj} \prod_{l \neq i, j \in N} (d_{jl} - w_{jl}a_l) \right| \;\forall i \in N \tag{4.55}$$

This change may cause the system to not converge, since e_i changes the convergence requirement from (4.38). If the system does converge, then the same ordering of the bounded price of ignorance can be seen to apply as in the average chasing case.

4.6 Conclusion

This chapter examined two classes of cognitive networks, the potential and quasi-concave classes, showing that both converge to NE action vectors when the cognitive networks contain certain properties. Under alignment with the cost functions, these NEs converge to either local-optima (for the potential class) and global-optima (for the quasi-concave class) of the network objectives. Specific real-world applications of these classes were introduced for each class. Finally, several properties of the quasi-concave class were identified with respect to the critical design decisions: they are unaffected by selfish behavior, resilient to forms of ignorant decision making, and degrade gracefully under less than full cognitive control. We will investigate the applications of these classes further in the next two chapters.

Chapter 5

Wireless Multicast with Cognitive Networks

In this chapter, the CN concept is applied to a multicast flow lifetime problem to illustrate several aspects of the CN operation and performance. First, we illustrate how this real-world problem fits into the CN framework. Next, the performance advantage of a CN approach to the problem is quantified. Finally, this CN is used to provide a case study for the critical design decisions.

5.1 The Cognitive Network

Multicast networks involve the communication from one source to many destinations. Many factors may affect a wireless multicast flow's lifetime. For instance, traffic congestion can cause timeouts in upper layer protocols, interference can cause loss of connectivity at the PHY layer, and mobility can cause unexpected disconnections in traffic routing. However, for mobile and portable devices, one of the chief factors in the lifetime of a flow is the utilization of the energy contained in the batteries of the mobile radios. Particularly for multi-hop wireless flows, the lifetime is limited by the radios whose battery fails first. This is the radio whose lifetime our CN attempts to maximize.

This work is not the first to investigate lifetime routing in wireless networks; a large body of work on power-efficient routing exists in the literature. Gupta's survey [122] provides an excellent comparison of several power-efficient multicast algorithms for omni-directional antennas. Weiselthier [123] has examined this problem using directional antennas. A complete review of the related literature and an investigation using Mixed Integer Linear Programs (MILPs) for determining the optimal lifetimes can be found in [124]. Although primarily designed to illustrate the critical design decisions, this work is the first to provide a distributed, CN approach to multicast lifetime routing that incorporates energy efficiency considerations, directional antennas, and a SINR sufficiency requirement. Furthermore, in Section 5.2, we are the first to use a MILP mathematical model to provide a true comparative measure of the scheme's effectiveness.

The CN approach presented here utilizes a SAN with three modifiable elements: the radio transmission power, antenna directionality and element routing tables. The end-to-end of maximizing the flow lifetime is accomplished through a cognitive process that changes the states of these modifiable elements. The cognitive process consists of multiple cognitive elements with local objectives that repeatedly observe their environment, examine the impact of possible action choices, and ultimately make decisions that improve their objective, and in turn, the end-to-end objective.

5.1.1 Problem Description

A wireless network is made up of a collection of network elements with varying energy capacity. Some elements may be battery powered, with limited capacity, while others may be less mobile, with large, high capacity batteries. The lifetime of a data path, however, is limited by the radio utilizing the largest fraction of its battery capacity. By minimizing the utilization of this bottleneck radio, the lifetime of the path can be maximized. Furthermore, we consider a network where radios are equipped with with directional antennas, which are useful to reduce interference, improve spatial multiplexing, and increase range.

We model a network consisting of a set of radios $N = \{1, 2, \ldots, n\}$, in which the objective is to create a maximum lifetime multicast tree between source s and destination set D. As described earlier, the CN controls three modifiable network parameters: the radio transmission power (contained in the elements of vector $\mathbf{pt}$), the antenna directionality (angles are contained in the elements of vector ϕ) and element routing tables (contained in each node of the multicast tree T). The states of the modifiable elements are part of the action set A, of which the action vector $\mathbf{a}$ contains the current state of each modifiable element.

In the model used here, the lifetime of a radio is inversely proportional to the utilization of the radio's battery,

$$\mu_i = \frac{pt_i}{ca_i} \tag{5.1}$$

where pt_i is radio i's transmission power and ca_i is the remaining energy capacity of its battery. The lifetime of a data path is limited by the radio utilizing the largest fraction of its battery capacity, so over the entire multicast tree T, the lifetime will be inversely proportional to the utilization of the max-utilization radio.

$$\mu_T = \max_{j \in T} \{\mu_j\} \tag{5.2}$$

The network consists of radios with fully directional antennas in receive mode[1] (each element transmits omnidirectionally and receives directionally) with a fixed beamwidth θ that can take on a boresight angle $\phi \in [0, 2\pi)$. Figure 5.1 illustrates the operation of an ad-hoc network with directional antennas in receive mode.

When radio i transmits, the signal experiences gain factor gb within the main beam of the antenna [126].

$$gb = \frac{2\pi}{\theta} \tag{5.3}$$

[1] An argument for using directional reception rather than for transmission or both can be found in [125].

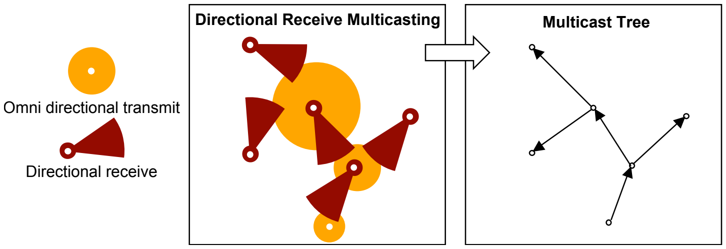


Figure 5.1: The directional receive multicast operation. The shaded areas extending from the radios represent regions of increased gain.

Some energy leaks outside the main beam in sidelobes. The fraction that ends up in the beam is $pct \in (0, 1)$ and the fraction outside the beam is $(1 - pct)$. We also consider a path loss attenuation factor, proportional to

$$gp_{ij} = \frac{1}{d(i,j)^{\alpha}} \tag{5.4}$$

where $d(i, j)$ is the euclidean distance between source i and destination j and α is the path loss exponent. Combining these gains and attenuations, the overall gain from a transmission by radio i received at radio j is

$$g_{ij}(\phi_j) = \begin{cases} gb \cdot gp_{ij} \cdot pct & \phi_j \in a(i,j) \pm \frac{\theta}{2} \\ gp_{ij} \cdot (1 - pct) & otherwise \end{cases} \tag{5.5}$$

where $a(i, j)$ is the angular function between radios i and j.

A radio j can correctly receive information from radio i if the power received from the desired transmitter is greater than all other power and noise received by some SINR factor. We define the vector **pr** to be the power received at every radio in the tree from their parent radio,

$$pr_j(pt_i, \phi_j) = pt_i \cdot g_{ij}(\phi_j) \tag{5.6}$$

There is an entry in this vector for every radio in the tree, with the exception of the source radio ($|pr| = |T| - 1$). We then define vector **no** to be the minimum required power to overcome the interference and noise received at every element,

$$no_j(\mathbf{pt}, \phi_j, T) = \left(\sum_{k \neq i} pr_k(pt_i, \phi_j) + \sigma_j \right) \gamma_j \tag{5.7}$$

where σ_j is the thermal noise and γ_j is the SINR requirement for a particular radio. The vectors **pr** and **no** combine to give the network constraint,

$$\mathbf{pr} - \mathbf{no} \geq \mathbf{0} \tag{5.8}$$

5.1.2 Cognitive Network Design

The CN framework encompasses a wide spectrum of possible implementations and solutions. This approach allows the framework to be a method for approaching problems in complex networks, rather than a specific solution. The framework sits on top of existing network layers, processes, and protocols, adjusting elements of the SAN to achieve an end-to-end goal. In this section we show how a CN that solves the multicast lifetime problem fits into the framework. We examine each layer, showing how the requirements layer provides goals to the cognitive elements, how the cognitive process performs the feedback loop, and identify the functionality of the SAN. The ideas in this section are illustrated in Figure 5.2.

The cognitive process consists of three cognitive elements that distribute the operation of the cognitive process both functionally and spatially: PowerControl, DirectionControl and RoutingControl. PowerControl adjusts the PHY transmission power (pt_i), DirectionControl adjusts the MAC spatial operation (ϕ_i), and RoutingControl adjusts the network layer's routing functionality (T).

The SAN network status sensors provide each cognitive element with partial-knowledge of the network. Battery utilization and routing tables are only reported within a radio's *k-hop neighborhood*. The k-hop neighborhood of a radio is defined to be every radio reachable in the

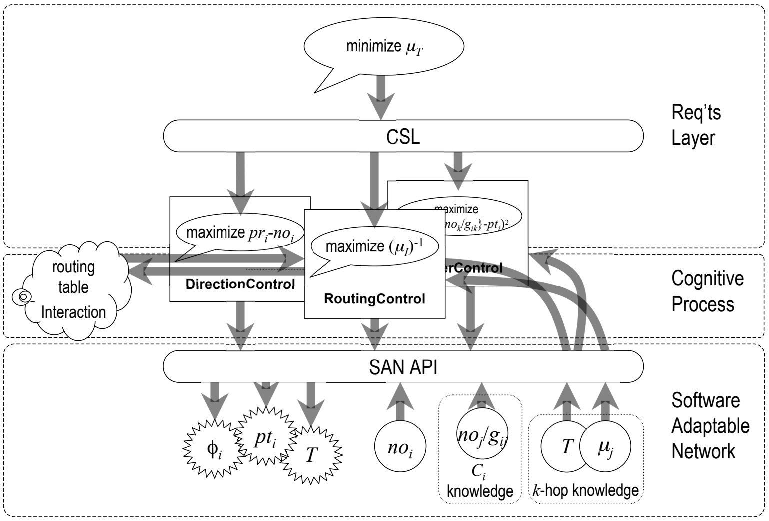


Figure 5.2: The components of the multicast lifetime cognitive network as they fit into the framework

routing tree via k hops, following the routing tree both up and down branches, as illustrated in Figure 5.3. The set of k-hop neighbors for radio i is represented by $\mathcal{N}_i^k$. The SAN also provides information about the required power needed to meet the SINR requirement of each of the next hop (child) radios in the multicast route. The set of child radios for radio are represented by $\mathcal{C}_i$.

Requirements Layer

The CN investigated here has a single objective optimization as its end-to-end goal. As such, the cost of an action vector is only dependent on the life-span of the multicast flow. The cost of a multicast flow is defined in (5.9), where the lifetime of a flow is increased as $C(\mathbf{a})$ is minimized.

$$C(\mathbf{a}) = \begin{cases} \mu_T & \mathbf{pr} - \mathbf{no} \geq \mathbf{0} \\ \infty & \text{otherwise} \end{cases} \tag{5.9}$$

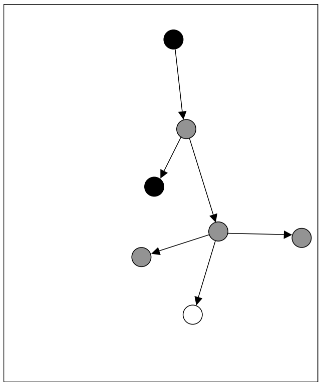


Figure 5.3: A sample k-hop neighborhood. The grey radios are in the 2-hop neighborhood for the white radio

Each of the modifiable elements affects the calculation of this cost: transmission power affects the lifetime directly; antenna orientation and routing table affect the lifetime indirectly by modifying the required transmission power.

The requirements layer transforms the end-to-end objective into a goal for each cognitive element through the CSL. Although these objectives are localized (each element only adapts a single modifiable element) the state of all modifiable elements affects the cognitive element's performance.

PowerControl's objective is to minimize the transmission power on every radio subject to the system constraint. This means that a radio will attempt to transmit at the minimum power that connects it to all of its children through the local control of pt_i. The objective can be represented by the utility function

$$u_i^{\mathrm{PC}}(\mathbf{a}) = -\left(\max_{k \in \mathcal{C}_i} \left\{ \frac{no_k}{g_{ik}} \right\} - pt_i \right)^2 \tag{5.10}$$

which is maximized when the transmitting radio has exactly the power needed to reach the child radio with the greatest noise and least gain factor. $\mathcal{C}_i$ is the set of child radios that receive from radio i in the multicast tree.

The objective of DirectionControl is to maximize the receiving radio's SINR by controlling the directional angle of the antenna beam ϕ_i locally at every antenna. One form that the utility can take is:

$$u_i^{\mathrm{DC}}(\mathbf{a}) = pr_i - no_i \tag{5.11}$$

By rotating the directional antenna, the radio can increase the gain from the parent radio, while attenuating interfering signals.

The objective of RoutingControl is to minimize each radio's battery utilization by manipulating the routing tree (T) used by the network. The utility can be expressed as:

$$u_i^{\mathrm{RC}}(\mathbf{a}) = \frac{1}{\mu_i} \tag{5.12}$$

By manipulating the children radios that it has to transmit to, radios can reduce their transmission power and battery utilization.

Cognitive Process

The cognitive process consists the three cognitive elements described above, each operating on every radio in the network. In this section, we discuss the strategies utilized by these elements to achieve the above objectives goals and identify the critical design decisions used by each cognitive element.

PowerControl PowerControl was first presented in chapter 4 as an application of the quasi-concave class with the end-to-end objective of minimizing the transmission power of each radio in the network. However, the PowerControl problem described before was point-to-point, with each transmitter connecting to a single receiver. The multicast problem can involve point-to-multipoint, with transmitters communicating to several child receivers. Assuming the routing tree and beam angles are fixed. the PowerControl utility function given by (4.26) can be re-written for the multicast requirement as

$$u_i(\mathbf{pt}) = -\left(\left(\sum_{k \neq i \in N} g_{kj} pt_k + \sigma_j\right)\gamma_j - g_{ij} pt_i\right)^2 ;\ j = \arg\max_{k \in \mathcal{C}_i} \left(\sum_{l \neq i \in N} g_{lk} pt_l + \sigma_k\right)\frac{\gamma_k}{g_{ik}} \quad (5.13)$$

When slightly re-formed (without any loss in generality) this objective becomes that shown in (5.10).

Proposition 5.1.1. *If A is a compact, convex subset of $\mathbb{R}^n_+$, $w_{ij} > 0$ for all i, j, $c_i > 0$ for all i, and $\alpha^t \equiv 1$ then the multicast PowerControl converges under Yates' to a PONE.*

Proof. Observe that u_i is is continuous in pt_i and $u_i^{max} = 0$ is constant with respect to $\mathbf{pt}$ for every $\mathbf{pt} \in A$. Expanding the noise and power received terms, the best response function $\mathbf{z}(\mathbf{pt})$ is given by:

$$\mathbf{z}(\mathbf{pt}) = \max_{k \in \mathcal{C}_i} \left\{ \left(\sum_{j \neq i \in N} g_{jk} pt_j + \sigma_k\right)\frac{\gamma_k}{g_{ik}} \right\}$$

We prove by going through each item of Yates' convergence (Section 4.3):

1. $\mathbf{z}(\mathbf{pt})$ is always positive because the gain terms are all positive ($g_{ij} > 0$ for all i and j) and thermal noise and SINR requirements are nonzero and positive ($\sigma > 0$, $\gamma > 0$).

2. Monotonicity is assured if $\mathbf{pt}' \succ \mathbf{pt}$ then $\mathbf{z}(\mathbf{pt}') \succ \mathbf{z}(\mathbf{pt})$, meaning that

$$\max_{k \in \mathcal{C}_i} \left\{ \left(\sum_{j \neq i \in N} g_{jk} pt'_j + \sigma_k\right)\frac{\gamma_k}{g_{ik}} \right\} > \max_{k \in \mathcal{C}_i} \left\{ \left(\sum_{j \neq i \in N} g_{jk} pt_j + \sigma_k\right)\frac{\gamma_k}{g_{ik}} \right\}$$

   for every $i \in N$. This can be proved by observing that since $g_{ij} > 0$, $\mathbf{z}(\mathbf{pt})$ is the sum of $n - 1$ monotonically non-decreasing linear functions. Thus increasing any pt_i increases all $z_j(\mathbf{pt})$. Thus $\mathbf{z}(\mathbf{pt}') \succ \mathbf{z}(\mathbf{pt})$ and the monotonicity requirement is met.

3. Scalability is assured if $\phi > 1$ and $\phi\mathbf{pt} \in A$ then $\phi\mathbf{z}(\mathbf{pt}) \succ \mathbf{z}(\phi\mathbf{pt})$

$$\phi \max_{k \in \mathcal{C}_i} \left\{ \left(\sum_{j \neq i \in N} g_{jk} pt_j + \sigma_k \right) \frac{\gamma_k}{g_{ik}} \right\} > \max_{k \in \mathcal{C}_i} \left\{ \left(\sum_{j \neq i \in N} g_{jk} \phi pt_j + \sigma_k \right) \frac{\gamma_k}{g_{ik}} \right\} \forall i \in N \tag{5.14}$$

   Assuming the thermal noise and SINR requirements are the same at every radio it is clear that

$$\arg\max_{k \in \mathcal{C}_i} \left\{ \left(\sum_{j \neq i \in N} g_{jk} pt_j + \sigma_k \right) \frac{\gamma_k}{g_{ik}} \right\} = \arg\max_{k \in \mathcal{C}_i} \left\{ \left(\phi \sum_{j \neq i \in N} g_{jk} pt_j + \sigma_k \right) \frac{\gamma_k}{g_{ik}} \right\}$$

   Thus (5.14) can be re-written as

$$\phi \frac{\gamma_k}{g_{ik}} \sum_{j \neq i \in N} g_{jk} pt_j + \phi \frac{\sigma_k \gamma_k}{g_{ik}} > \phi \frac{\gamma_k}{g_{ik}} \sum_{j \neq i \in N} g_{jk} pt_j + \frac{\sigma_k \gamma_k}{g_{ik}} \, \forall i \in N$$

   and, since $\sigma_k > 0$, inspection shows that scalability holds.

4. We can promise that a pure strategy NE $\hat{\mathbf{a}}$ exits in A. Since A is a compact, convex subset of $\mathbb{R}^n_+$ and u_i is quasi-concave with respect to a_i, from the Glicksberg-Fan fixed point theorem [117, 118], Γ^z has a fixed point and thus a pure strategy NE in A.

5. From assumption.

□

Since this shows that the multicast PowerControl is also a member of the quasi-concave class of problems, we utilize the selfish RELAX strategy to move the transmission power of the elements of a tree to a minimum but sufficient NE power state (referred to as $\hat{\mathbf{pt}}$) for a given tree structure. This means that parent radio i will iteratively increase or decrease its transmission power to the least amount needed to overcome the interference and noise observed by all children. Algorithm 1 describes RELAX more formally.

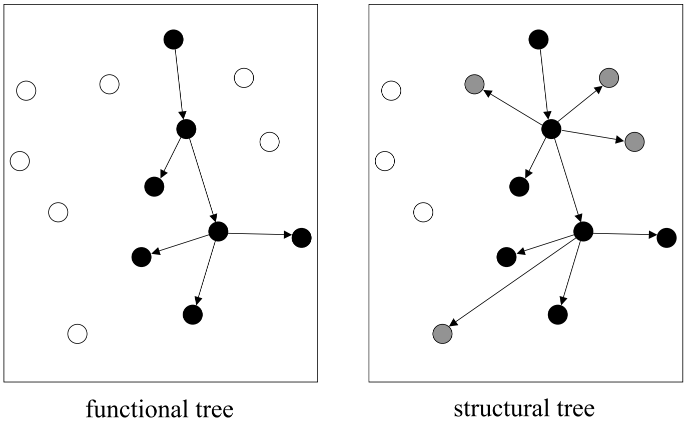
Algorithm 1 $\text{Relax}(\mathbf{pt}, \phi, T) \rightarrow \hat{\mathbf{pt}}$

```
while $\hat{\mathbf{pt}}$ is not a NE do
    for $i = 1 \ldots n$ do
        $pt_i = \max_{j \in \mathcal{C}_i}\{\frac{no_j}{g_{ij}}\}$
    end for
end while
```

Relax fits the the definition of selfishness, as it exclusively chooses future power levels that increase an elements' utility, assuming that all other elements continue to transmit at the same level. However, this selfishness is complementary to the end-to-end goal, as increasing $u_i(\boldsymbol{a})$ can only decrease $C(\boldsymbol{a})$. PowerControl operates under ignorance in that it is not explicitly aware of the states of all other radios in the network. An individual radio need only be aware of the set $C_i \in T$ and the values of no_j/g_{ij} for each radio in the set. When every radio contains a PowerControl cognitive element, transmission power is under full control.

DirectionControl The second cognitive element's behavior is DirectionControl. DirectionControl moves the directional antenna to the orientation that maximizes the received SINR from a node's parent node. There are several direction-finding algorithms in the literature [127] and DirectionControl can implement one of these. If a node is a part of the multicast routing tree, it directs its antenna such that the power received from the parent is maximized with respect to the amount of interference and noise. If a node is not part of the multicast tree, it directs the antenna towards any source from which it can receive with the greatest SINR. For clarity, we will delineate these two tree structures: the first, called the *functional tree*, consists of just elements in the multicast routing tree and the second, called the *structural tree* includes every element in the system that can receive a signal that meets the SINR requirement. These two trees are illustrated in Figure 5.4.

Based on the objective given in (5.11), the network is playing a dummy game. The choice of antenna orientation of one radio has no effect on the utility of any other DirectionControl


functional tree

structural tree

Figure 5.4: The difference between the structural and functional tree. The white radios are not part of the multicast tree. Black radios are part of the functional and structural trees; grey radios are listener radios that are only part of the structural tree.

Algorithm 2 BeamFind($\mathbf{pt}, \phi, T$) → $\hat{\phi}$

```
1: for $i = 1 \ldots n$ do
2:     choose $\phi_i$ so that $pr_i - no_i$ is maximized
3: end for
```

element. In [10], Neel observes that a dummy game is an EPG (and from the relationships between potential games described in Section 4.1, also an OPG). Thus any selfish strategy will lead to a NE. Due to the simplicity of a dummy game, it is also apparent that this equilibrium will be PO. Algorithm 2 describes the shell of a selfish strategy called BeamFind, in which line 2 assumes a direction finding algorithm such as those described in [127] to find antenna orientation that maximizes the SINR.

Like Relax, BeamFind exhibits selfishness, only selecting actions that improve the cognitive elements' objective. Also similarly to PowerControl, by maximizing the local SINR, DirectionControl is maximizing the signal quality for the individual radio, also promoting the end-to-end goal by reducing the amount of power needed to communicate between radios. Furthermore, PowerControl operates under partial knowledge, having only to know its local SINR. PowerControl has full control over the antenna orientation modifiable element.

RoutingControl RoutingControl attempts to minimize the utilization of the radio batteries by approximating a Steiner tree for the utilization metric. RoutingControl uses the ChildSwitch strategy described in algorithm 3. ChildSwitch begins by determining if it is operating on the *max-utilization* radio (the radio with maximum battery utilization) of its k-hop neighborhood, by comparing its battery utilization against every k-hop neighbor's battery utilization. If it is, the radio becomes the *control-radio* and takes control over the routing tables of every element in the k-hop neighborhood. It then identifies which of the children radios in the functional tree requires the greatest amount of power to reach (the *max-power child*). The control-radio then attempts to detach the max-power child from itself and re-attach it as the child of another radio (by changing the routing table of a k-hop neighbor so that it becomes the new parent) in the k-hop neighborhood, in order to reduce the k-hop neighborhood's maximum utilization.

Valid choices for a new parent for the max-power child include all radios in the k-hop neighborhood of the structural tree, except for children of the max-power child. By selecting parent radios from the structural tree, new radios in the network can be brought sensibly into the functional tree. After assignment, ChildSwitch waits until Relax converges and DirectionControl selects the correct beam angle. When Relax converges, ChildSwitch on the control-radio compares the utilization of all radios in the k-hop neighborhood against its initial utilization. The process is then repeated for the remaining valid radios, with the control-radio remembering the best (minimum) max-utilization configuration, and upon completion setting the routing table to this configuration.

This process repeats indefinitely until the max-utilization control radios are no longer able to move their max-power children to configurations that lower the max-utilization radio of their k-hop neighborhood. In a synchronous network, in which only one RoutingControl control-radio performs ChildSwitch at a time, the network will (except in rare cases) converge to a single set of max-utilization radios.

Theoretically, there may be cases in which $k < dia(T)$ (where $dia(T)$ is the diameter of

Algorithm 3 ChildSwitch$(\mathbf{pt}, \phi, T) \rightarrow (\hat{\mathbf{pt}}, \hat{\phi}, T')$

```
if $\mu_i = \max_{n \in \mathcal{N}_i^k}\{\mu_n\}$ then {is on a max-util. node}
    $\mu_{minmax} = \mu_i$ {record the config. as the min-max}
    $minmax = i$
    $j = \mathrm{argmax}_{c \in C_i}\{no_c\}$ {record the max-power child}
    for $n \in \mathcal{N}_i^k \quad n \neq j;\ n \notin B_j$ do {every valid neighbor}
        $C_n = C_n \cup \{j\}$ {add max-power child}
        $\hat{\phi} = \mathsf{DirectionControl}(\phi)$
        $\hat{\mathbf{pt}} = \textsc{Relax}(\mathbf{pt})$
        $\mu_{max} = \mathrm{argmax}_{n \in \mathcal{N}_i^k}\{\mu_n\}$ {record max-util.}
        if $\mu_{max} < \mu_{minmax}$ then {max-util. is least}
            $\mu_{minmax} = \mu_{max}$ {record it as the min-max}
            $minmax = max$
        end if
        $C_n = C_n \backslash \{j\}$ {remove max-power child}
    end for
    $C_{minmax} = C_{minmax} \cup \{j\}$ {change to min-max config.}
end if
```

the multicast tree) that permit the cycling of the topology between more than one set of maximum utilization radios. As an example, this occurs when ChildSwitch reduces the maximum utilization radio in one k-hop neighborhood, causing an increase in max-utilization in a non-overlapping k-hop neighborhood elsewhere in the tree. This unintended interaction can occur when some radios increase their transmission power (although increase them less than the max-utilization) during PowerControl. The network cycles when the non-overlapping k-hop neighborhood reduces its max-utilization in a similar manner, causing an increase the utilization in the original k-hop neighborhood. The cases where this phenomenon occurs are rare, requiring the perfect amount of interaction between unrelated k-hop neighborhoods, something that most multicast tree algorithms prevent by sanely choosing trees with spatially separate (and thus free from such interaction) branches. Practically, this effect was never observed in any simulation performed in Section 5.2.

A more realistic cause of cycling is the lack of synchronous RoutingControl operation. As long as only one control-radio is making changes at a time, the memory and observations of the control-radio will be accurate and repeatable. In an asynchronous network (in which more than control-radio performs RoutingControl at the same time) there is the possibility of cycling, in which the network bounces between more than one set of max-utilization radios. Changing the frequency with which control-radios attempt to perform this procedure may reduce the probability of asynchronous operation occurring. In this case, the expected time between ChildSwitch invocations would be much greater than the running time of ChildSwitch itself. Alternatively, the SAN could be used to synchronize these operations.

ChildSwitch is a heuristic strategy, and as described above its performance is not necessarily optimal or correct. In the average case however, ChildSwitch will decrease the maximum utilization of the network. Section 5.1.3 will describe the performance of the algorithm in the average case. The performance of the algorithm is not boundable; it is possible to create scenarios in which RoutingControl will find the optimal max-lifetime flow or scenarios in which RoutingControl will find an tree with lifetime arbitrarily worse than the optimal.

The operation of ChildSwitch is similar to edge-swapping heuristics. The movement of a max-cost child to a k-hop neighbor swaps a high cost connection (edge) for a lower cost connection. However, most edge swapping algorithms are not designed to operate under partial knowledge and assume fixed edge weights. ChildSwitch operates under k-hop knowledge and edge weights that change from iteration to iteration due to changing interference levels.

Note that ChildSwitch follows the feedback loop process outlined in Section 2.2.1. The cognitive element on the control-radio begins by *observing* the max-power child. It then *orients* itself by moving the max-power child to each of the k-hop neighbors. Based on the information gained from this orientation, it *decides* which parent radio results in the minimum max-utilization, and *acts* by permanently changing the routing tree so that the max-power child is underneath this parent. This action changes the environment. After this change, the control-radios are re-determined and the process is repeated.

Unlike the other two cognitive elements, RoutingControl exhibits the feature of altruism, since elements (in order to decrease the end-to-end objective cost) must allow a control-radio to move a max-power child to them, which may increase their utilization. Because of the k-hop neighborhood, RoutingControl operates with a degree of ignorance. The signal it observes, $y_i = \{\mu_j | j \in \mathcal{N}_i^k\}$ is imperfect because it is missing information about utilization from those radios outside the k-hop neighborhood. RoutingControl operates under full control, with control over the routing tree at every radio.

Software Adaptable Network

The SAN provides an interface to the three modifiable network elements and the status of the network. The reported status is the local noise, maximum transmission power required to reach its children, k-hop battery utilization and k-hop routing tree.

The required transmission power, battery utilization of child radios and routing tree status can be discovered and reported via a variable power handshaking scheme. In a synchronous manner, radios one by one send a Hello message addressed to all children. Each child responds with an Ack message to the parent. The parent then decreases its transmission power and sends a new Hello message until it fails to receive an Ack from some child. The parent radio then stops decreasing its power and returns to the previous power level, which is the maximum transmission power required to reach all its children. These Hello and Ack messages can also transfer information about each radio's battery utilization and the routing tree within the k-hop neighborhood. In contrast to these non-local measurements, the amount of local noise can be calculated through the local SINR measurement.

5.1.3 Results

We develop a simulation to determine the effectiveness of this CN. The simulation was written in Matlab, and consisted of nodes placed with a uniform random distribution in a square 2-D map with density 0.1 nodes/unit^2. There is a single source node and a variable number of receivers. The beam width θ is 30 degrees, the path loss exponent α is 2, and 30% of the transmitted power is assumed to leak out through sidelobes $(1 - pct)$. Each wireless node was given a battery with a random capacity (ca_i) uniformly distributed between 0 and 300 units of energy. The SINR sufficiency requirement is set to 1, meaning that the received power must be greater than the noise and interference to satisfy (5.8).

The normalized lifetime of a path is calculated as the ratio of the lifetime obtained by the CN to the optimal lifetime for the same set of source/destinations, capacities, and node positions. The optimal lifetime was determined using Wood's MILP, the full formulation of which can be found in [125]. Knowing the optimal solution is useful, since it allows a true "apples-to-apples" comparison between different scenarios, resulting in an accurate gauge of how effective the CN is.

Underlying the CN, one of two different generic multicast routing algorithms was used. The first, Greedy, uses a greedy algorithm to create the multicast tree. Greedy forms the multicast tree from the source node, adding minimum utilization nodes until a spanning tree has been formed. Utilization is estimated for every pair of nodes as the ratio of the (non-interference) transmission power required to reach each node to the node's battery capacity. Greedy then prunes off branches until it has the minimum tree required to reach every destination. The other multicast algorithm used is Star, which implements a one-hop broadcast star from the source to every destination.

For a given scenario (consisting of node count, location and battery capacity) both Greedy or Star were run, individually. The resultant tree topology and parent/child information from each algorithm were handed to Relax and DirectionControl, which determine pt_i and ϕ_i, maximizing the lifetime for this route. The full cognitive process, including RoutingControl was then run on the route determined by Greedy and Star until it converged to a single set of max-utilization nodes. The lifetime of the resultant tree was then calculated. At least 100 simulation runs were performed per scenario. Finally, both the non-cognitive and cognitive lifetimes for a scenario were compared against the optimal lifetime obtained from

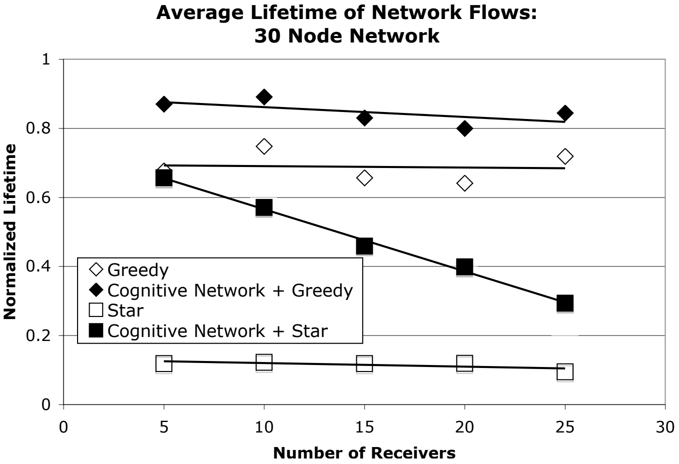


Figure 5.5: Normalized lifetimes for Star and Greedy routing algorithms, before and after cognitive process adaptations.

Routing Algorithm	Number Rcvrs	Path Length Max	Path Length Avg	Radio Degree Max	Radio Degree Avg
Star	10	+336%	+108%	-49%	+56%
	20	+332%	+92%	-45%	+16%
	30	+282%	+71%	-37%	+6%
	40	+305%	+61%	-33%	+2%
Greedy	10	+11%	+7%	-8%	+5%
	20	+9%	+7%	-6%	+3%
	30	+11%	+8%	-5%	+2%
	40	+10%	+7%	-12%	+1%

Table 5.1: Percent change in graph parameters for 50 radio network

the MILP, providing a normalized lifetime in (0, 1], where 1 represents an optimal lifetime for that particular scenario.

Figure 5.6 contains the mean normalized lifetimes for selected 10 radio network scenarios with varying k-hop neighborhood size. This data confirms that the simplicity of Star alone does indeed lead to sub-optimal performance, with at worst case less than 40% of the average lifetime of Greedy. However, it also confirms that the CN can make a significant improvement on the average lifetime of the flow by using a 1-hop neighborhood – over 125% improvement in the Star case. The Greedy algorithm alone achieves much longer lifetimes, but the CN is still able to improve it by 5-15%. In both routing algorithms, lifetimes

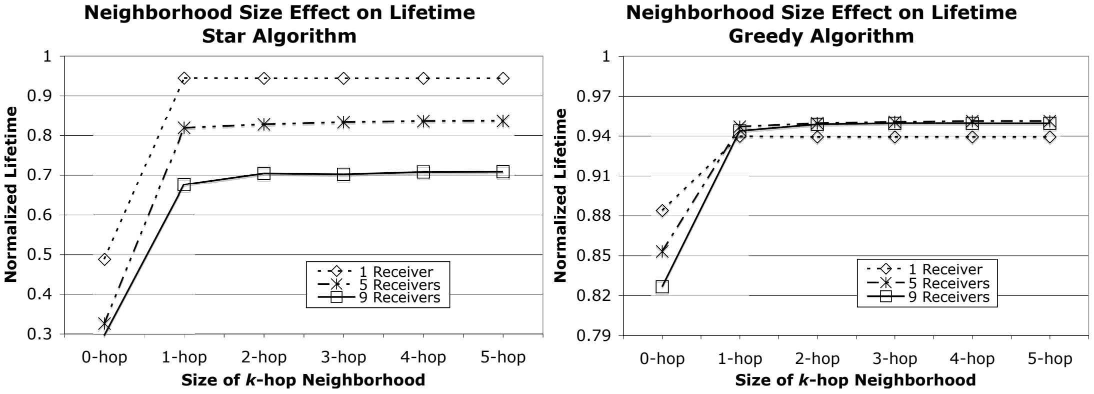


Figure 5.6: The mean normalized lifetimes of the CN for various 10 radio network k-hop neighborhoods. 0-hop neighborhoods represent the performance of the underlying routing algorithm (i.e., a "non-CN" approach). Note the differing scales of the y-axis.

remained steady or decreased as the number of multicast receivers increased. The CN was able to improve the lifetime of the connection for all receiver counts and neighborhood sizes. More interestingly, the data shows that neighborhoods beyond 1-hop have negligible impact on the lifetime of the flow.

The size of the network was increased to 30 radios, with 5, 10, 15, 20 or 25 receivers per network. All parameters, including the radio density were maintained from the 10 radio scenarios. Figure 5.5, which shows the average normalized lifetimes for the underlying routing algorithms and the improved normalized lifetimes that the cognitive process provides. Because of the results in Figure 5.6, only the 1-hop neighborhood was used. These results are consistent with the graph metrics contained in Table 5.1, which shows the percent change in path length and radio degree on the multicast topology graph after the cognitive process improves the networks.

In these scenarios, the lifetime of the flow without the cognitive process and only Greedy or Star is constant independent of the number of receivers. Furthermore, the average lifetime of the CNs operating on the Greedy algorithm are statistically identical over the range of receiver counts considered, overlapping at the 95% confidence interval. The cognitive process shows a decreasing ability to increase the flow lifetime as the number of receivers increases, which is attributable to the increasing complexity of the tree structure required to reach these additional receivers.

So far, this CN has illustrated the framework and provided a performance justification. In the next section, this CN will be used to examine and quantify the critical properties and the price of the features discussed in the beginning of the chapter.

5.2 Effect of critical properties

In Chapter 3, the price of a design a feature metric was introduced and defined. In this section, we analyze the expected price of a feature for the three critical design decisions: selfishness, ignorance, and control.

5.2.1 Price of Selfishness

Although PowerControl and DirectionControl exhibit selfishness, the cognitive process as a whole exhibits altruism due to RoutingControl. To modify RoutingControl's ChildSwitch strategy so that it operates selfishly ($c =$ selfish), if the act of taking on a new child increases a k-hop neighbor's utilization, that neighbor will not accept the child.

Figure 5.7 shows the expected price of selfishness for networks utilizing both Star and Greedy. For the Star routing algorithm, the price of selfishness is defined by the performance of Star itself. All Star trees consist of a topology in which a single source radio transmits to all receivers. Each of these receivers are passive, listening but not transmitting, meaning that any change of topology will require one of these receiver radios to transmit, increasing their utilization. Since selfishness dictates that a radio will only take on a new child if it requires the same or lower transmission power, selfishness does not allow for any modification of Star's initial tree. The cognitive process provides decreasing improvement

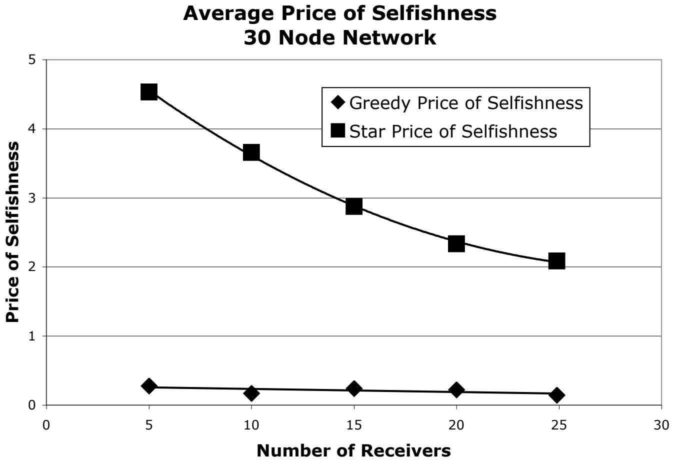


Figure 5.7: Expected price of selfishness of 30 radio network, for Star and Greedy routing algorithms.

as the number of receivers increases (as is apparent from Figure 5.5), and thus the expected price of selfishness also decreases.

In contrast, the expected price of selfishness for the Greedy case is statistically constant (average lifetimes have overlapping 95% confidence intervals). This is because even when acting selfishly, there are ways that RoutingControl can increase the lifetime of the routes selected by Greedy. Since they are utilizing directional antennas, it is possible for receiving radios to meet the SINR requirement from more than one transmitter without either transmitter increasing its utilization. The expected price of selfishness in the Greedy case is about 0.25, meaning that the lifetime of the flow is 25% shorter when acting selfishly instead of altruistically, regardless of the number of receivers.

5.2.2 Price of Ignorance

The signal that PowerControl and DirectionControl observe to learn the transmission powers of the other radios is the aggregate sum of the other radios' attenuated powers and thermal noise. This measurement contains imperfect information due to indistinguishability, since there are many transmission powers that can lead to the same observation. Fortunately, this ignorance has no effect on these cognitive elements' decisions. In contrast, as discussed in Section 5.1, RoutingControl is ignorant due to missing information, which can affect the element decisions.

This standard feature (c = ignorance) can be changed by giving the RoutingControl elements on the control-radios knowledge of the utilization of the entire network, meaning a parent-child modification will only be kept if it decreases the maximum utilization of the entire network. In many cases, this encourages modifications that directly move the network towards the maximum lifetime action vector. However, this may trap the network in local minima by excluding all decisions that do not immediately help the network lifetime.

Figure 5.8 shows the expected price of ignorance for the Star and Greedy cases. Interestingly, the expected price of ignorance is less than 0 (meaning that ignorance is *better* than knowledge) for the five-receiver case in the Star network. Full knowledge can get stuck in some local maximas, partial knowledge causes the network to make decisions that may be poor in the short term but turn out to be beneficial in the long term. Particularly for small network sizes, the relative large amount of network knowledge provided from the 1-hop neighborhood allows for it to arrive at better network performance than full knowledge. However, the expected price of ignorance increases (and is positive) as the number of receivers increases. More significantly, the expected price of ignorance is relatively small and never as high as the expected price of selfishness.

5.2.3 Price of Partial-Control

This cognitive process effectively has control over every modifiable element in the network. Although the number of modifiable elements being controlled in any given iteration is dependent on the size of the k-hop neighborhood, all modifiable elements in the network can be controlled by the cognitive process. The price of partial control can be calculated by changing the standard feature of full control to partial-control (c = partial-control) by designating a certain fraction of the radios to be uncontrollable by the cognitive process.

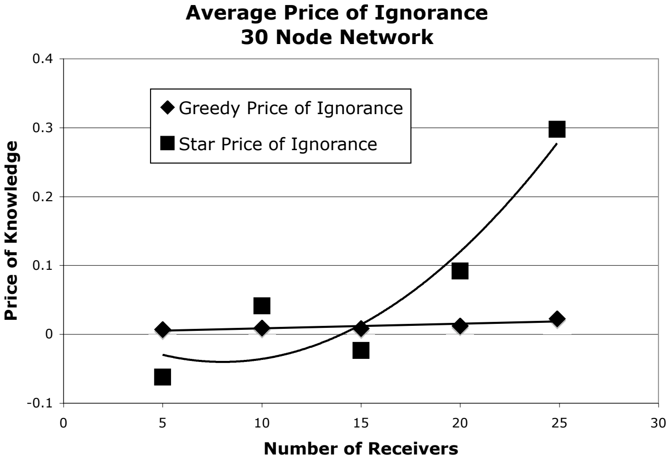


Figure 5.8: Expected price of ignorance for 30 radio network, with both Star and Greedy routing algorithms.

However, without the control that cognitive elements PowerControl and DirectionControl provide, selected trees may be infeasible. For this reason, PowerControl and DirectionControl and the control they have over the transmission power and beam angles are maintained at all elements. Instead, RoutingControl, and the control it provides over the routing tables, is removed from a fraction of the radios. When RoutingControl is not operating on a radio, it is assumed that the radio acts “dumb,” only making routing adjustments that come from the network stack – either Greedy or Star – and cannot act as a control-radio, max-power child or new parent to a max-power child.

The results in Figure 5.9 show the expected price of partial control for a network that has a 1-hop neighborhood. Four different levels of cognitive control for each of the five different receiver counts were examined, spanning the spectrum from slightly reduced to minimal cognitive control. The data here shows that regardless of the number of receivers, the expected price of partial control increased as the amount of cognitive control is decreased. Additionally, the expected price of 20%-control for Star is statistically identical to the expected price of selfishness, meaning that for networks with less than 20% cognitive control, the cognitive process cannot make improvements on the multicast trees that Star has selected.

This case study has illustrated a particular CN and revealed several interesting design guide-

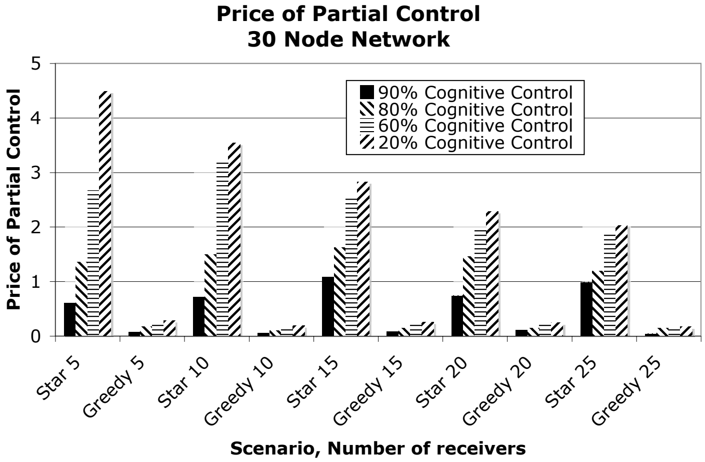


Figure 5.9: Expected prices of partial control (90%, 80%, 60% and 20% cognitive-control) for 30 radio network, with both Star and Greedy routing algorithms.

lines. For instance, the designer may decide that the price of ignorance in RoutingControl is insignificant especially when considering that giving RoutingControl more than 1-hop knowledge may require significant communication overhead. In comparison, the price of partial control is quite high, making a case for near-full cognitive control. The designer may also find justification for the altruistic operation of RoutingControl in the high price of selfishness, particularly for Star.

5.3 Conclusion

Although computer networks are becoming increasingly ubiquitous, the ability to manage and operate them is not becoming increasingly easier. This section outlined a CN approach to a difficult multicast problem. This approach quantitatively improved the lifetime of the multicast flow over non-cognitive heuristic methods. Furthermore, the CN provided a basis for illustrating the framework and the critical design decisions. In particular, the areas of control and selfishness were shown to have the greatest effect on the network performance.

Chapter 6

Topology Control With Cognitive Networks

Wireless networks, in the most general sense, are unstructured. With increasing programmability, radios are able to autonomously adapt to their environment, setting transmission power and selecting their frequency of operation. Topology control attempts to harness this programmability to build structure into the wireless network. Our work addresses two questions related to distributed topology control: in selecting their operating parameters, should radios be programmed to optimize their own "selfish" utilities or a network-wide objective function?; and, how much knowledge about network state must be available to each radio to enable it to make adaptation decisions that are efficient in a network-wide sense?

Traditionally, the field of topology control has examined power control problems that disregard spectral efficiency or vice-versa. Unlike most of the literature on topology control, we examine the objectives of lifetime and spectral efficiency jointly. Our goal is to establish a distributed framework for minimizing maximum transmission power and spectral footprint, all the while achieving interference-free connectivity.

This chapter examines several topological issues from a CN perspective. Using a selfish, autonomous framework, it examines the formation of power and spectrum efficient topologies under static and dynamic conditions. It also examines the performance impact of partial knowledge about the network and compares this against the cost of gathering and communicating the information.

6.1 Problem Background

To address these issues of spectrum and power control, we propose a two-phased CN approach in which the cognitive elements are distributed on each radio of the network. We use two potential class cognitive elements first introduced in Section 4.2: TopoPowerControl, which controls the transmission power of a radio, and TopoChannelControl, which chooses the transmission channel for a radio. Our goal is to establish a distributed framework for interference-free communication between selfish radios. Each of the cognitive elements observes the network conditions and then selfishly chooses either the reduced power level that still maintains topological connectivity (in the case of TopoPowerControl) or a channel that allows interference-free connectivity with all desired receivers (in the case of TopoChannelControl). Pursuing these selfish objectives, the network reaches a topology state that minimizes both the maximum transmission power and the number of orthogonal channels required to achieve interference-free connections. A graphical representation of how these cognitive elements fit into the CN framework is presented in Figure 6.1.

6.1.1 Related Work

Reducing spectrum usage in a network of radios has typically been examined in conjunction with interference avoidance. In general, interference avoidance has led to three viewpoints: those of the radio, the topology and the network. The radio viewpoint attempts to find interference-minimizing channels at the link level, without considering network formation or topology; [89] presents an example of this using cognitive radios. The network perspective

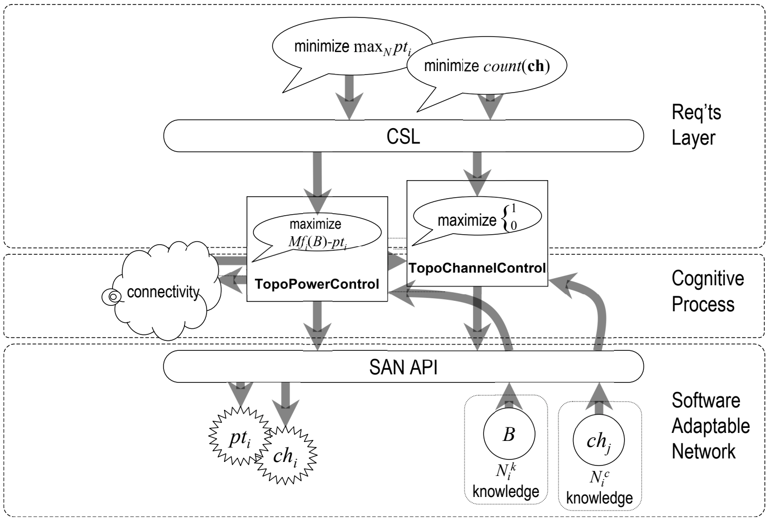


Figure 6.1: The components of the topology control cognitive network as they fit into the framework.

assumes an already existing topology over which the radios are attempting to communicate, with channel assignments made in either an on-demand or pre-assigned manner; [128] presents a network-perspective scheme that combines routing with channel assignment. Our CN approach takes neither the radio viewpoint, which lacks enough scope to make network-aware decisions, nor the network viewpoint, which does not fully leverage the flexibility of the wireless medium; but instead utilizes the topological viewpoint.

Previous work on interference avoidance topologies can be broken into two assumptions: the network has power control or the network has channel control. Burkhart [129] pioneered the power approach, assigning to each connection a weight equal to the number of radios the connection interferes with. This is used in the Min-Max Link Interference with property $\mathcal{P}$ (MM-LIP), Minimize the Average Interference Cost while Preserving Connectivity (MAICPC) and Interference Minimum Spanning Tree (IMST) [130] algorithms. Another power approach uses a radio interference function, in which the interference contribution of a radio is the maximum interference of all connections incident upon it. This is adopted in the Min-Max Node Interference with property $\mathcal{P}$ (MMNIP) [130] and the Low Interference-Load Topology (LILT) [131] algorithms. Alternatively, the Average Path Interference (API) [132] approach trims high-interference, redundant edges from the Gabriel graph. The channel control approach assumes the connectivity of the network is fixed and that two radios can only communicate if they share a common channel, of which there are fewer available than needed; this is illustrated by Connected Low Interference Channel Assignment (CLICA) [133], a heuristic approach, and Subramanian's Tabu-search based algorithm [134].

For assigning non-conflicting channels to power-induced topologies, many channel assignment problems are modeled as the distance-2 coloring problem. This model describes the connections in the topology as a graph and assigns different colors (channels) to vertices of distance one and two from each other. This approach was used in [135] for (r, s)-civilized graphs using a centralized, cooperative approach. Generalized bounds on the capacities induced from channel assignments of this model can be found in [136]. We do not use the distance-2 coloring problem as a model since it overstates the scope of the intereference problem directional connections in topologies. Furthermore, our strategies do not place limits on the underlying connectivity graph.

Another model used for assigning non-conflicting channels is the edge-coloring problem. This model assigns different colors to the edges of the connection graph so that no two edges that join at a vertex are colored the same. A simple distributed, probabilistically close to optimal coloring algorithm is given in [137]. For the MAC scheduling problem (closely related to the channel assignment problem), [138] provides an edge coloring scheme and describes its bounded performance. We do not use the edge coloring problem as a model since it does not capture the broadcast nature of the radios well.

To achieve energy efficiency, topology control through transmit power control generally utilizes a single shared channel and assumes a MAC for temporal separation of interfering transmissions. Many excellent topology control algorithms have been proposed to create power-efficient topologies; for a recent survey, see [139]. Most works deal with network dynamics indirectly, by developing local, fault-tolerant algorithms such as [140, 141]. Alternately, probabilistic models are developed and are often argued for as being robust to mobility [142]. General results are achieved in [143], with the authors describing computational complexity bounds on minimizing the maximum or total power used in the topology. Centralized, cooperative techniques are presented for approximating the solutions to these problems. In [144], Li examines an approach to maintaining connectivity in a topology by restricting the maximum geometric angle between connections, allowing only local knowledge to be used. However, the radios need to be aware of the directions from which their neighbors are transmitting.

Despite these claims, a thorough evaluation and analysis of the impact of partial information on network optimality is still missing from most work in the literature. Furthermore, most of these algorithms assume total cooperation amongst radios, which collectively set their transmission power level so as to achieve a network-level goal. The cone-based approach given by Li in [144] has a local-knowledge algorithm for topology construction, but makes no guarantees for the performance of the algorithm. In [145], the authors examining routing under partial knowledge of the topology, determining an optimal geographic range in which the topology should be known to make energy efficient routing decisions.

The use and analysis of selfishness for topology control was examined in [146], in which the authors investigate a rudimentary definition of a topology and determine whether or not certain general sub-problems will arrive at a NE and what the worst-case performance of this equilibrium is under selfish behaviors. Similar work by [147] also examines the price of selfishness, but here it is examined for Peer-to-Peer (P2P) networks and this work makes many assumptions about the possible connectivity and local objectives that do not translate to wireless, ad-hoc networks.

Unlike previous works, we consider radios that can regulate both power and channel selections. Additionally, each cognitive element selfishly adapts to the perceived network conditions to maximize its own performance. We assume that knowledge of the network state is local, and that knowledge of another radio's state is a function of the connectivity distance between them. We ensure that such independent, self-motivated, local-knowledge element-level adaptations align well with the network-level goals. The novelty of our work lies in unifying distributed, local, and autonomous characteristics of self-organization in topology control through learning and reasoning.

6.1.2 Problem Description

In order to formally describe our two end-to-end objectives, we begin by describing the radio and topological model.

We use the radio propagation model described in Section 4.2, in which the transmission range of radios are modeled as disc-like. The topology created by these connections is contained in $G = (N, E)$ where N is the set of radios and E is a set of directed arcs that represent the unidirectional connections. Equation 6.1 shows the set of connections: a connection exists if the transmission power (pt_i) is greater than the thermal noise (σ) and SINR requirement (γ) at the receiving radio, given the gain (loss) factor between the two (contained in the symmetric matrix $[g_{ij}]$). We only consider thermal noise and not interference because of the orthogonal nature of the to-be selected channels.

$$E = \left\{ e_{ij} \mid pt_i \geq \frac{\sigma\gamma}{g_{ij}} \right\} \tag{6.1}$$

The value of $\sigma\gamma/g_{ij}$ will be referred to as $\omega(\overline{ij})$, which is the transmission power required to form a connection from radio i to radio j. In a slight abuse of notation, we will sometimes use $\mathbf{p}$ to represent G (noting that $\mathbf{p}$ induces G).

We assume that topological connectivity comes from bi-directed connections, which occur when there exist connections between radios in both directions. The set of bi-directed connections consists of members of E that have their reverse also in E. Because our model acknowledges only bi-directional links, G is connected if and only if there exists a bi-directed path—a collection of contiguous bi-directional links—between every radio pair $i, j \in N$.

This leads to the system constraint for the power induced topology: these bi-directed connections must produce a bi-directed path between every radio and every other radio in the network.

$$f_i(\mathbf{pt}) - n = 0 \, \forall i \in N \tag{6.2}$$

In this equation $\mathbf{f} : A \rightarrow \mathbb{Z}_+^n$ is a vector function where $f_i(\mathbf{pt})$ returns the number of radios reachable from i as described in (4.2). We express the objective of minimizing the maximum transmission power in the network as the cost function:

$$C(\mathbf{pt}) = \max_{i \in N} pt_i \tag{6.3}$$

The spectral usage objective can be described using the channel conflict graph U, which relationship to G is defined in (4.1). The conflict graph consists of undirected edges drawn between radios that cannot share a common channel without causing conflict.

We extend the static network model to include dynamic changes through the addition or removal of a radio from the network. Under discrete updates, mobility will appear to the network as radios appearing and disappearing. This also represents dynamic changes that do not involve radio mobility. In particular, in Wireless Local Area Networks (WLANs) radios often drop in or out as users turn their machines on or off. Wireless sensor networks also consist of radios that wake up or go to sleep periodically, adding or removing themselves to/from the network.

To prevent radios from interfering, the channel chosen by the cognitive element on radio i must be different than the channels chosen by its conflicting neighbors. If the vector $\mathbf{ch}$ contains the channel indices of the network radios, such that ch_i is the channel index used by radio i, then the following expresses the system constraint that no radio use the same channel index as its conflict neighbors:

$$\forall i \in N, \; |ch_i - ch_j| > 0 \;\text{ if } \exists e_{ij} \in U \tag{6.4}$$

Under this system constraint, the second objective is to minimize the spectral footprint of the topology. This is contained in the number of channels utilized, and can be expressed by the cost function:

$$C(\mathbf{ch}) = count(\mathbf{ch}) \tag{6.5}$$

where the function $count : \mathbb{Z}^n \to \mathbb{Z}_+$ returns the number of unique channel indices in vector $\mathbf{ch}$.

The cost function for this CN is difficult to quantify since there are two noncommensurate metrics in (6.3) and (6.5), Watts and channels. We leave the problem open by not specifying a relationship between the two and expressing the cost as a 2-tuple, $(Watts, channels)$.

To achieve the dual objectives of power and spectrum control, the CN operates as two sequential games. The first game models power control, where cognitive elements attempt to minimize their transmit power level and maintain network connectivity. The output of this is a power-efficient topology, which is fed into the second game, where transmission channels that do not have conflicts are selected. Both of these games can be envisioned as performing a cognitive cycle: the elements select (and possibly revise) their optimum settings based on the perceived topology state; these revised actions induce a change in topology configuration, either in the connectivity or in the channel index profile; the modified topology affects the utility of individual elements, which in turn update their power or channel settings, and the cycle starts all over again.

We examine these cognitive elements and their strategies with an eye towards how knowledge affects performance. Knowledge is information combined with context. In a cognitive process, knowledge is used to appropriately apply learned data. For many problems, without knowledge, even the cleverest cognitive process will not have any way to select one decision over another, while in contrast, full knowledge can potentially allow a cognitive process to select the best decision.

Power and spectrum control are especially appropriate for cognition and operation under partial knowledge. Minimizing the number of channels, even under full knowledge, bears many similarity to the well-known graph coloring problem, which is NP-hard to solve for arbitrary graphs. Hard problems such as these are well-suited for cognition, since there are no polynomial time solutions to the problem and all current solutions are heuristic in nature. Cognition may provide an edge in approximating the optimal solution. Minimizing the maximum transmission is closely related to the formation of the Minimum Spanning Tree (MST), which requires full knowledge to guarantee the correct formation. Thus a cognitive process that operates under some degree of ignorance in achieving this objective is a useful improvement. In previous work, both of these problems have been investigated using cooperative, centralized approaches. These are in contrast with the approach used here, in which the objectives are broken down into distributed, selfish cognitive elements.

6.2 Power Objective

We examine the problem of power control under two environments: static and dynamic. For dynamic environments, we examine the problem under full and partial network knowledge. Under full network knowledge, we present strategies that determine the optimal topology with respect to the power objective. We then investigate the effectiveness of these strategies under partial knowledge.

6.2.1 Static Networks

Static networks are networks of radios that do not change during the time frame required for the power settings to be adapted to a network state. Dynamic networks, on the other hand, are networks that change in a way that force the power settings to be re-adapted.

Full Knowledge

The δ-Improvement Algorithm (DIA) developed in [148] and described in Algorithm 4 (from a network-wide perspective) is an example of a full knowledge, selfish strategy guaranteed to converge to a lifetime-maximizing topology.

The input to DIA is a max-power topology with every radio transmitting at an initial power $pt^{(0)} \in \bigcap_{i \in N}[0, pt_i^{\max}]$ that ensures network connectivity [149]. Each cognitive element executes a selfish better-response dynamic by reducing its power level one step lower than its previous level if the change improves (4.2). To enable such discrete adaptations, it is sufficient to search over the *modified action set*:

$$\tilde{A}_i^{\mathrm{PC}} = \left\{pt^{(0)}, pt^{(1)}, pt^{(2)}, \ldots\right\} \tag{6.6}$$

such that at most one connection is dropped in the network when the power vector is adapted from $\mathbf{pt}^{(m)}$ to $\mathbf{pt}^{(m+1)}$. Note that every element shares this common action set and $\tilde{A}_i^{\mathrm{PC}}$ is an ordered set, with $pt^{(m+1)} < pt^{(m)}\ \forall m$. One way to construct $\tilde{A}_i^{\mathrm{PC}}$ is to initialize all radios to $pt^{(0)}$ and decrement power in a predefined step size δ such that $pt^{(m+1)} = pt^{(m)} - \delta\ \forall m \geq 0$. A value of δ that satisfies this requirement can be defined by:

$$\delta = \min_{e,f;e \neq f} |\omega(e) - \omega(f)| \tag{6.7}$$

Since A is a compact set, such a δ exists for random networks.

Note that DIA is a full knowledge strategy because the action set is constructed from information on the minimum power required by every radio to reach every other radio, and action choices are chosen synchronously. Synchronous action selection requires every cognitive element to know the network's current action choice. Furthermore, the utility function of the cognitive elements requires the evaluation of the f_i function, which requires global knowledge to determine how connected the network is.

Algorithm 4 DIA$(G) \rightarrow (G_{dia}, \hat{\mathbf{pt}})$

```
m = 0
$\hat{pt}_i = pt^{(m)} \in \tilde{A}_i^{\text{PC}}\ \forall i \in N$
while $\hat{\mathbf{pt}}$ is not a NE do
    $m = m + 1$
    for all $i \in N$ do
        choose $pt_i = pt^{(m)} \in \tilde{A}_i^{\text{PC}}$
        $\hat{pt}_i = \arg\max_{pt'_i \in \{pt_i, \hat{pt}_i\}} u_i^{\text{PC}}\left(pt'_i, \hat{pt}_{-i}\right)$
    end for
end while
```

DIA converges to a topology wherein the maximum power of a radio is minimized. Observe that in each iteration the max-power connection in the topology is disconnected. Such an iterative process eventually converges to a topology that contains the shortest $n-1$ edges that ensure connectivity. In some sense, the NE is a superset of the MST connections, containing the bi-directional links that make up the MST and the additional extraneous links induced by the wireless broadcast property of the medium. We first present the following three lemmas which are essential in proving this main result. The first two lemmas were proven by Komali in [148].

Definition 6.2.1 (power Minimum Spanning Tree)**.** *A topology is a power Minimum Spanning Tree (pMST) if its bi-directed connections form only the MST and all other connections in the graph are induced from the wireless broadcast property of the medium.*

Lemma 6.2.1. *DIA converges to a NE that preserves connectivity of G_{max}.*

Proof. From theorem 4.2.1 we know that TopoPowerControl is an potential class CN of the OPG type. From [108], it follows that since DIA is a selfish algorithm (better response), DIA will converge to a NE. However, we are interested in only those NE that preserve connectivity in the final topology.

The input to the DIA is the topology G_{max}, with every node transmitting at pt_i^{max}. The better response for each node is to reduce its transmission power (and improve its utility) to a value pt_i so that the resulting topology is remains connected. We prove this by contradiction.

Suppose node i improves its utility at $pt'_i < pt_i$, given pt_{-i}, and the network is not connected. This implies that $u_i\left(pt'_i, pt_{-i}\right) = M \cdot k_i - pt'_i > M \cdot n - pt_i$, where $k_i < n$, the total number of nodes in the network. This implies, $M \cdot (n - k_i) < pt_i - pt'_i$, an impossible inequality, because the term on the left hand side is larger than pt_i^{max} and the term on the right hand side is smaller than pt_i^{max}. Thus, in every round, the topology is always connected. □

Lemma 6.2.2. *Consider the* TopoPowerControl *problem where nodes employ the DIA. Starting with an initial topology G_{max} induced by the power vector $\mathbf{pt}^{(0)}$, the algorithm converges to a subgraph, G_{dia}, of the G_{pmst}.*

Proof. Proof is by induction. For the ease of presentation, we suppose, without loss of generality, that G_{max} is a complete network and $\omega(\overline{ij})$ is equal to the euclidean distance between the corresponding nodes, given by the distance function $d(i,j)$. (Note that this means g_{ij} is a symmetric matrix).

Consider a G_{max} comprising of 3 nodes: a, b, c; suppose $d(a,b) > d(a,c) > d_(b,c)$ be the relationship between euclidean distances. Nodes start at power level $pt_i = d(a,b)$ for all i in N and keep decreasing their power in steps of δ, until $pt_i = d(a,c)$. At this point, nodes a and c will not reduce their power any further; otherwise, the network would disconnect and the nodes' payoff would decrease[1]. Because $pt_a = d(a,c)$ and $d(a,b) > d(a,c)$, link $\overline{ab}$ is severed as a result. Thus, the DIA algorithm converges to a topology containing links $\overline{ac}$ and $\overline{bc}$, the shortest two bidirectional links needed to connect the three nodes.

Now consider a fully connected topology with 4 nodes: a, b, c and d; let $d(d,a) > d(d,c) > d(d,b) > d(a,b) > d(a,c) > d(b,c)$. All nodes keep decreasing their power from $d(d,a)$ until they reach $d(d,b)$. Node d now has only a single link, $\overline{db}$, that is bidirectional. The problem then reduces to a 3 node topology as before. Thus, the algorithm converges to a topology containing the three shortest bi-directional links $\overline{ac}$,$\overline{bc}$ and $\overline{bd}$ (and possibly some induced unidirectional links as well).

[1] According to the argument in Lemma 6.2.1, which also applies for a DIA dynamic, network connectivity is preserved at every stage of the game.

The above line of reasoning can be generalized to any arbitrary network of size n. Therefore, the algorithm always hits a state that consists of the shortest $n-1$ bidirectional links needed to maintain connectivity. Note that, at this point the network is at G_{pmst} by definition.

If G_{pmst} contains a bidirected cycle (a cycle with all bidirectional links), at least one node in the cycle may still reduce its power level further and still maintain connectivity. Otherwise, G_{pmst} contains exactly all the bidirectional links of MST. In either case, the steady state topology G_{dia} is a subgraph of G_{pmst} (the subgraph may not be proper). This completes the proof. □

Lemma 6.2.3. *G_{pmst} minimizes maximum power of any given node in the network.*

Proof. The main idea behind the proof is the fact that MST minimizes the maximum edge-weight of the network. We show this by contradiction.

Let us assume, on the contrary, that there exists another tree T_{other} different from T_{mst} that minimizes the maximum edge-weight. Let

$$e_{other}^{max} = \arg\max_{ij \in T_{other}} \omega\left(\overline{ij}\right)$$

and

$$e_{mst}^{max} = \arg\max_{i,j \in T_{mst}} \omega\left(\overline{ij}\right)$$

By our contradiction, $\omega\left(e_{other}^{max}\right) < \omega\left(e_{mst}^{max}\right)$. Introduce a cut—and partition the nodes into two sets N_1 and N_2—in T_{mst} by removing e_{mst}^{max} from the graph. Since T_{other} is a tree, we can find an edge $\tilde{e} \in G_{other}$ to join N_1 and N_2 and create an new tree $\tilde{T}_{other}$. Because e_{other}^{max} is the edge in T_{other} with the maximum weight, we have $\omega\left(\tilde{e}\right) \leq \omega\left(e_{other}^{max}\right) < \omega\left(e_{mst}^{max}\right)$. Since $\tilde{T}_{other}$ is essentially created from T_{mst},

$$\sum_{e \in \tilde{T}_{other}} \omega(e) < \sum_{e \in T_{mst}} \omega(e)$$

we obtain a contradiction. Therefore, T_{mst} is indeed the tree with the minimum maximum edge-weight.

The edge, e_{mst}^{max}, with the maximum weight determines the node with the maximum power. The powers in G_{pmst} are dictated by the bi-directed connections that make of the MST, not the induced edges. Therefore, it follows that G_{pmst} minimizes the maximum power of any node in the network. □

Using the above lemmas, the following main theorem is an immediate consequence.

Theorem 6.2.4. *DIA converges to topology G_{dia} that minimizes the maximum power of any given node.*

Proof. From Definition 6.2.1 we know that G_{pmst} contains the MST and all the additional induced connections. Because none of the induced connections increase the maximum connection power of the graph, G_{pmst} preserves Lemma 6.2.3. From Lemma 6.2.2, the steady state topology G_{dia} is a subgraph of G_{pmst}; therefore, every connection in G_{dia} is contained in G_{pmst}. It follows immediately that Lemma 6.2.3 still holds for G_{dia}. Hence, the result follows. □

By iterating through the cognitive cycle, DIA captures reasoning and a rudimentary form of learning. After every iteration, the action choices in $\tilde{A}_i^{\text{PC}}$ are non-increasing, as there is no rational reason to consider previous, higher powers. In this sense, the cognitive elements learn from past actions and eliminate unproductive actions from their search space. Furthermore, the radios act upon perceived network conditions and make decisions to revise their transmit power in order to improve their performance – this represents a level of reasoning.

Partial Knowledge

Construction of a MST cannot be accomplished in a purely localized manner [150]. This means that if radios have only partial knowledge about the transmission powers required to make connections with other radios in the network, they cannot guarantee the correct selection of connections that make up the MST. This section presents a strategy derived from DIA that constructs minimal power topologies under partial knowledge.

To quantify knowledge, we introduce the idea of a k-hop neighborhood. The k-hop neighborhood in a topology is defined as the set of radios reachable within k hops via bi-directional connections. The members of each successive k-hop neighbor of i can be described recursively:

$$\mathcal{N}_i^k = \begin{cases} \{i\} & k = 0 \\ \mathcal{N}_i^{k-1} \cup \left\{ j \mid e_{jl}, e_{lj} \in E,\ l \in \mathcal{N}_i^{k-1}, j \neq i \right\} & k > 0 \end{cases} \tag{6.8}$$

We assume that radios have k-hop knowledge, which means the radios have full knowledge of the network state in their k-hop neighborhood and no more. Furthermore, we assume that k-hop knowledge is not transitive, radios do not share information beyond their k-hop neighborhood, and there is no passive learning, meaning that radios only use information explicitly shared with them and do not utilize information overheard in the wireless medium. These assumptions allow k-hop knowledge to be an experimental parameter that can be tuned to study the role of partial knowledge in network design. Note that if $k \geq dia(G)$ (where $dia(G)$ is the diameter of the topology, the maximum number of hops between any two radios in the network) then the network is said to be operating under full-knowledge since every radio's k-hop neighborhood includes the full network. For all $k < dia(G)$, there may be some degree of partial-knowledge, in the sense that some or all radios may be unaware of a portion of the network state. Also note that the fraction of the total network that the cognitive elements are aware of is a function of their k-hop knowledge and the topological connectivity; we explicitly examine this relationship in Section 6.4.

As described in the previous subsection, the original DIA algorithm is global in its scope; it utilizes full knowledge of the network to determine the connectivity of the network, define the possible transmission powers and synchronize the power selection across the network. Local-DIA is described in Algorithm 5 from a single cognitive element perspective. It has been generalized from DIA so that it can operate without global knowledge of network connectivity, required transmission power, or synchronization state. As in the case of DIA, Local-DIA operates on an action set wherein each radio searches for the optimum action over those power values that correspond to the power thresholds for each reachable neighbor. Under full knowledge, Local-DIA becomes functionally the same as DIA. Generalizing (6.6), we define the action set for each radio as follows:

$$\bar{A}_i^{\text{PC}} = \left\{ p_i^{\max} = pt_i^{(0)}, pt_i^{(1)}, pt_i^{(2)}, \ldots, \right\} \tag{6.9}$$

such that one connection is dropped by radio i when the power is adapted from $pt_i^{(m)}$ to $pt_i^{(m+1)}$. One way to construct $\bar{A}_i^{\text{PC}}$ is to initialize radio i to $pt_i^{\max}$ and decrement its power by a variable step size δ_i^m. Because $pt_i^{(m)}$ represent the connection power thresholds, the first step-size is given by $\delta_i^0 = pt_i^{\max} - \max \left\{ \omega(\overline{ij}) \mid \omega(\overline{ij}) < pt_i^{\max} \text{ for all } j \right\}$ and subsequent step-sizes are given by $\delta_i^m = pt_i^{(m)} - pt_i^{(m+1)}\ \forall m > 0$. Such step-sizes exist for each m for the reasons given before.

In some sense, the action set (6.9) specifies how a local topology control game might evolve: each individual radio starts by transmitting at its maximum power, reasons according to a better response process and chooses the next power level in the set if it observes an improved utility and informs other radios in its k-hop neighborhood.

To achieve its power control objective, the cognitive element still employs (4.2) as its utility function to determine its local connectivity. To accommodate the lack of full-knowledge, however, f_i is now the number of the radios in the k-hop neighborhood of radio i. This change allows radios to determine their local connectivity under partial knowledge.

Local-DIA being a better response algorithm, each radio only chooses powers that increase their utility; hence, radios never choose a power level that reduces the size of their set of k-hop neighbors.

Algorithm 5 Local-DIA$(G) \to \hat{p}_i$

```
$m = 0$
$\hat{pt}_i = pt_i^{(m)} \in \bar{A}_i^{\text{PC}}\ \forall i \in N$
while $\hat{\mathbf{pt}}$ is not a NE do
    $m = m + 1$
    choose $pt_i = pt_i^{(m)} \in \bar{A}_i^{\text{PC}}$
    $\hat{pt}_i = \arg\max_{pt'_i \in \{pt_i, \hat{pt}_i\}} u_i^{\text{PC}}\left(pt'_i, \hat{pt}_{-i}\right)$
end while
```

Lemma 6.2.5. *In* Local-DIA *the dominant strategy for every radio is to preserve its k-hop neighborhood.*

Proof. We prove this by contradiction. Suppose radio i reduces its power level from pt_i to q_i to reduce its k-hop neighborhood (from, say, $\mathcal{N}_i^k$ to $\mathcal{N}_i'^k$) and increases its utility. This implies that $u_i\left(q_i, pt_{-i}\right) = M_i|\mathcal{N}_i'^k| - q_i > M_i|\mathcal{N}_i^k| - pt_i$, where $|\mathcal{N}_i'^k| < |\mathcal{N}_i^k|$. This implies, $M_i\left(|\mathcal{N}_i^k| - |\mathcal{N}_i'^k|\right) < pt_i - q_i$, an impossible inequality, because the term on the left hand side is larger than $pt_i^{\max}$ and the term on the right hand side is smaller than $pt_i^{\max}$. □

Recall that under partial knowledge radios cannot ensure synchronization and do not have knowledge of the values of $\omega(\overline{ij})$ for the network. Due to this reason, Local-DIA, under partial knowledge, does not necessarily converge to an optimal network power configuration. While each radio maintains its k-hop neighborhood, according to Lemma 6.2.5, its decision may reduce another radio's k-hop neighbor set; for an illustration, see Figure 6.2. The following theorem ensures that, following Local-DIA, every NE still preserves the overall network connectivity.

Definition 6.2.2 (Partitioned Topology)**.** *A partitioned topology G is a topology that has 2 or more partitions such that there does not exist an bi-directed connection between any of the partitions.*

Theorem 6.2.6. *If initial topology G is connected, then* Local-DIA *converges to a NE that is also connected.*

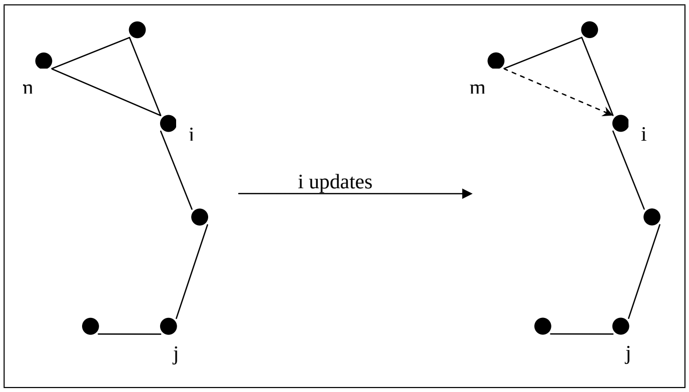


Figure 6.2: Illustrating the Local-DIA process: For $k = 3$, radio i can maintain connectivity with radio m at reduced power level, but this affects radio j, which loses m from its 3-hop neighborhood.

Proof. A connected network can be disconnected in two ways:

1. if a radio (say i) disconnects itself from another radio, while executing Local-DIA or,

2. if radio i disconnects two previously connected radios, say j and m, in the process of reducing its power during the course of Local-DIA.

We know that the case 1 violates Lemma 6.2.5. The latter case is not possible unless j and m are connected to each other through i, in which case i will not lose connection with either j or m by virtue of Lemma 6.2.5. Thus, in either case, the network will remain connected. □

The sub-optimality of the resultant Local-DIA topology is exacerbated as the amount of knowledge is decreased. The resultant topologies are over-connected under partial knowledge, given that radios will not remove any connection that decreases the size of their k-hop neighborhood. Furthermore, unlike DIA, which synchronously changes power to remove connections, there is a “first mover advantage” inherent in Local-DIA. The first radio to act has the most actions available to it; subsequent radios have their action spaces reduced by previous radios’ action choices. Although Theorem 6.2.4 of DIA no longer holds (*i.e.*

minimizing the maximum transmission power) in case of Local-DIA, Local-DIA still generates topologies that are Pareto-efficient.

Theorem 6.2.7. *Every NE topology that emerges from* Local-DIA *is PO.*

Proof. Firstly, no node can reduce its power any lower; otherwise the network would be disconnected and hence violate Theorem 6.2.6. Secondly, no m-node (where $m \geq 2$) reduction in power levels can preserve the network connectivity either. This is because, if some node reduces its power (and therefore, disconnects the network), some other node must increase its power to re-connect the network. It follows that the new configuration is not pareto-optimal. □

6.2.2 Dynamic Networks

Dynamic networks, as described earlier, are networks that change in a way that forces the power settings to be re-adapted and the topology re-formed. This change can take on many forms. Radios can move or the environment can change, which affects the required power needed to connect radios. We model all these as changes resulting from the addition or removal of radios from the network.

Adding Radios

When adding radios to the topology, to maintain the G_{dia} topology, the new radio needs to connect into the existing topology and the existing topology needs to add and remove bi-directional connections such that the MST properties of G_{dia} are retained.

This strategy for maintaining G_{dia} after the addition of a radio can be summarized as follows: the new radio forms a least-power bi-directional connection with some existing radio in the topology. Under full knowledge, each radio transmits at the current maximum power in the network and begins DIA (without loss of generality, this maximum power can be denoted by

$pt^{(0)}$ and we assume such a common $pt^{(0)}$ exists). Under partial knowledge, we approximate this by having each radio with k-hop knowledge of the new connection set its power to the current maximum power in its k-hop neighborhood, then beginning Local-DIA.

Local-DIA-Add, described in Algorithm 6 (from a radio perspective), contains two operations that can be triggered. First, "Hello" is triggered when a new radio, x, joins and attempts to connect into the topology. The radio closest to x (this is denoted by radio i in Line 2) responds by forming a bi-directional connection with it, then notifies its k-hop neighbors to "Local Restart." This second operation instructs each k-hop neighbor to increase power to the maximum transmission power in its k-hop neighborhood, and then run Local-DIA.

Algorithm 6 Local-DIA-Add(x) → $\hat{pt}_i$

```
1: while ⟨Hello|$x$⟩ do
2:     $pt_x = \min_{y \in N}\{\omega(\overline{yx})\}; i = \arg\min_{y \in N}\{\omega(\overline{yx})\}$
3:     $pt_i = \max\{\omega(\overline{ix}), pt_i\}$
4:     Local Restart→ $\mathcal{N}_i^k$
5: end while
6: while ⟨Local Restart⟩ do
7:     $pt_i = \max_{j \in \mathcal{N}_i^k} pt_j$
8:     $\hat{pt}_i$ =Local-DIA($\mathbf{pt}$)
9: end while
```

We now prove that under full knowledge, Local-DIA-Add restores G_{dia}. We begin with a lemma that identifies the bi-directional connections that make up a MST:

Lemma 6.2.8. *Cut radio set N into two subsets M and O such that $M \subset N$, $O = N \backslash M$. Let F be the set of all possible connections between M and O. If $e = \arg\min_{f \in F} \omega(f)$ then e is part of the MST.*

Proof. Assume MST T does not contain e. Let $e = \overline{ij}$, with i in M and j in O. This means i and j must be connected in T via some other bi-directed path. This path will include some connection f that connects M to O. The tree $U = T \cup \{e\} \setminus \{f\}$ is also a spanning tree,

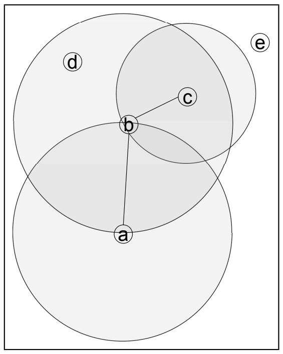


Figure 6.3: Radios a, b and c represent an existing G_{dia} topology. Radios d and e represent the case where a radio is added within and outside the current transmission ranges, respectively.

and its sum power is smaller than T, since $\omega(e) < \omega(f)$. Thus T is not a MST, and we have proven the lemma by contradiction. □

As discussed earlier, from the information exchanged over the course of LOCAL-DIA, the radio closest to the new radio correctly reasons to elect itself to increase its power and re-connect the topology. To verify this, we examine the impact of adding a radio on the maximum transmission power of the redefined topology, using the above lemma. Adding a new radio gives rise to one of two scenarios: the added radio falls inside the region covered by the transmission ranges of the existing radios, or the added radio falls outside this region. These two cases are illustrated in Figure 6.3. We first examine the former case:

Lemma 6.2.9. *If radio i is the least transmission power radio to radio x and $\omega(\overline{xi}) \leq \max_{e \in G_{dia}} \omega(e)$, then the maximum transmission power in the new topology G'_{dia} is less than or equal to that in the initial G_{dia} topology:*

$$\max_{e \in G'_{dia}} \omega(e) \leq \max_{e \in G_{dia}} \omega(e)$$

Proof. From Lemma 6.2.8, in a MST, radio x will be connected via bi-directional connection $\overline{xi}$, which requires less power than all other bi-directional connections from x. From our assumptions, $\omega(\overline{xi}) \leq \max_{e \in G_{mst}} \omega(e)$, so this has not increased the transmission power in G'_{dia}.

We now examine the remaining bi-directional connections in G'_{dia}. Let E be the set of all possible connections in G and let E' be the set of all possible connections in G' after adding radio x. Note that $E' \supset E$, meaning we have not removed any possible connections, only added possible connections. Returning to the notation of Lemma 6.2.8, for all cutting subsets of $N \cup \{x\}$, the set of possible connections $F' \supset F$. Thus

$$\min_{f \in F'} \omega(f) \leq \min_{f \in F} \omega(f)$$

and we have proven the addition of x will not increase the maximum connection power in the network. □

Next we examine the case when radio x falls outside the region covered by the maximum transmission range of the radios in the existing topology.

Lemma 6.2.10. *If radio i is the least transmission power radio to radio x and $\omega(\overline{xi}) > \max_{e \in G_{dia}} \omega(e)$, then the maximum power connections in the new topology G'_{dia} is the connection between x and i:*

$$\omega(\overline{xi}) = \max_{e \in G'_{dia}} \omega(e)$$

Proof. From the assumption that $\omega(\overline{xi}) > \max_{e \in G_{mst}} \omega(e)$ and Lemma 6.2.8 we see that all cutting subsets of $N \cup \{x\}$ that include x (except that with only x) will not select a connection originating from x, since it is not the minimum power connection between the cut. Thus adding radio x will not change the connections in the topology, with the exception of the min-power connection between x and i. □

Following lemmas 6.2.9 and 6.2.10, Local-DIA-Add ensures that, in the event a new radio joins the network, optimal topology G_{dia} can still be re-constructed with enough network awareness (*i.e.* no edge in G_{dia} is ever removed by Local-DIA-Add). Using these two lemmas, the following theorem is an immediate consequence, and establishes the correctness of Local-DIA-Add under full knowledge.

Theorem 6.2.11. *Under full knowledge,* Local-DIA-Add *re-converges to the G_{dia} topology.*

Proof. Lemmas 6.2.9 and 6.2.10 prove that regardless of the amount of power required to connect radio x, Local-DIA-Add will set all radios to the maximum required connection power, $\max_{e \in G_{dia} \cup \{\overline{xi}\}} \omega(e)$. Theorem 6.2.4 proves that this connected topology will converge to the G'_{dia} topology. □

Removing Radios

The other network dynamic we consider is the case where radios leave the network. The removal of a radio from the network can potentially split the existing connected topology into multiple partitions.

Full Knowledge Under full knowledge, a MST construction algorithm can be used to minimize the maximum transmission power and reconnect the network. This is accomplished by treating each partition as a "meta" radio, consisting of all the radios in a partition. The required connection powers between any two partitions then becomes the minimum $\omega(\overline{ij})$ between all radios i in the first partition and radios j in the second.

The resultant topology has several interesting properties as compared to the initial topology:

Lemma 6.2.12. *If $\mathcal{N}_i^k$ is the set of k-hop neighbors in the G_{dia} topology for radio i before the removal of the radio, $\mathcal{N}_i'^k$ is the set of neighbors in the G'_{dia} topology for radio i after the removal of the radio, and x is the removed radio, then no neighbors will be removed besides x in the topology transformation,* i.e.*:*

$$\mathcal{N}_i^k \backslash \{x\} \subseteq \mathcal{N}_i'^k \; \forall i \in N$$

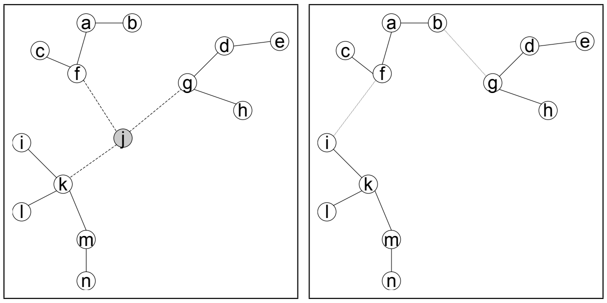
Proof. When removing radio x, the set of all possible connections in the network changes from E to E'. It is easy to see that $E' \subset E$. Therefore from Lemma 6.2.8, for every i and j not equal to x, if $\overline{ij}, \overline{ji} \in G'_{dia}$, then $\overline{ij}, \overline{ji} \in G_{dia}$ and all neighbors are retained by the radios in topology G' with the exception of x. □

Theorem 6.2.13. *If the removal of radio x creates n partitions $\{X_1, X_2, \ldots, X_n\}$ then the new G'_{dia} topology contains all bi-directional connections (except those including x) in the original topology G_{dia} plus the minimum weight bi-directional connections between the partitions.*

Proof. From Lemma 6.2.12, we see all bi-directional connections that do not include x are present in G'_{dia}. Furthermore, from Lemma 6.2.8, it is apparent that the minimum weight bi-directional connections between X_m and $N \backslash X_m$ are part of the MST and thus part of G'_{dia}. □

Partial Knowledge Unfortunately, under partial knowledge it is not possible to utilize a MST algorithm. The remove case does not have the same foundational knowledge that the add case does: the knowledge that the network is fully connected with the exception of the added radio. The remove case can create partitions up to the degree of the removed radio. Without full knowledge of the required power for all possible connections in the network and the members of all partitions, it is not possible to guarantee that all radios know when the topology is fully connected and what optimal connections to use.

As an example, see Figure 6.4. In this figure, radios experience the loss of radio j. To correctly form the G_{dia} topology, the partitions should use connections $\overline{if}$, $\overline{fi}$ and $\overline{gb}$, $\overline{bg}$. Under 3-hop knowledge, radios in the partitions do not know these are the correct connections, nor do they know that these connections reconnect the topology. For instance, radio k initially had radio g in its 3-hop neighborhood and under this reconnection does not. In fact, under this reconnection no radio $\{i, k, l, m, n\}$ in partition has all of their initial 3-hop neighbors in their new 3-hop neighborhood. Radios will not know the topology is reconnected unless



(a) Before removal of radio (b) After removal of radio

Figure 6.4: Reconstructing G_{dia} topology, after removing a radio from the network

all initial k-hop neighbors (with the exception of the radio that was removed) become k-hop neighbors again.

With these limitations in mind, we develop a localized strategy, called Local-DIA-Remove that captures the properties of the full knowledge topology, not removing any connections or k-hop neighbors from the partitioned topology when reconstructing. This is described in Algorithm 7 from the radio perspective, using $\mathcal{N}_i^k$ to denote the original k-hop neighborhood (before the radio removal) and $\mathcal{N}_i'^k$ to denote the current k-hop neighborhood (after the radio removal).

Algorithm 7 Local-DIA-Remove$(x, m) \rightarrow \hat{pt}_i$

```
1: $K = \mathcal{N}_i^k \backslash \left(\mathcal{N}_i'^k \cup \{x\}\right)$
2: **while** $K \nsubseteq \mathcal{N}_i'^k$ **do**
3:     $m = m - 1$
4:     $pt_i = pt_i^m \in \bar{A}_i^{pc}$
5: **end while**
6: $\hat{p}_i =$Local-DIA$(\mathbf{p})$
```

In Local-DIA-Remove, each radio sequentially increases its transmission power one level higher (as specified by line 4) in $\bar{A}_i^{pc}$ until its k-hop neighborhood is recovered (as specified by the while loop in line 2). These power increases will create uni-directional connections, eventually to be complemented with their reverse, creating bi-directional connections that add at least one more radio into the k-hop neighborhood. By following a random ordering, eventually all radios will have recovered their original k-hop neighborhood. In the expected sense, this sequential stepwise strategy will reconnect the topology with fewer connections.

Local-DIA-Remove is not guaranteed to converge to the G_{dia} topology, and in most cases it will not. Unlike Local-DIA-Add, which resets the entire network to a common power level under full knowledge, Local-DIA-Remove resets the network to differing power levels and because of the order of action updates, it may not re-connect the least-power connections between partitions. This means even under full knowledge, Local-DIA will operate in an unsynchronized manner.

6.3 Spectrum Objective

We now investigate a selfish, best-response strategy for TopoChannelControl. This strategy is examined first with respect to static networks, and then under dynamic changes.

6.3.1 Static Networks

The objective of TopoChannelControl to minimize the spectrum usage of the topology by activating the least number of orthogonal channels is the same as that of graph coloring and there are many different heuristic strategies in the literature. In order to minimize the total number of colors, a possible best response strategy is for each randomly ordered radio (the ordering is called permutation π) to choose the lowest non-conflicting channel index; we call this strategy Local-RS, as it is a localized version of the Random Sequential coloring algorithm described in [151]. A formal description of the operation of Local-RS is contained in Algorithm 8 (from the network perspective).

Local-RS works by randomly assigning a backoff to each radio in the network within a fixed window. When the radio's backoff ends, it selects the lowest channel that does not conflict with its neighbors to transmit on. These backoff periods induce an ordering to the

Algorithm 8 Local-RS$(U, \pi) \rightarrow \hat{\mathbf{ch}}$

```
while $\hat{\mathbf{ch}}$ is not a NE do
    for $i$ in $\pi$ do
        $\hat{ch_i} = \min\{A_i^{\text{cc}} \setminus \{ch_j | e_{ij} \in U\}\}$
        $ch_i = \hat{ch_i}$
    end for
end while
```

coloring problem, represented by π in Algorithm 8. This repeats until every channel has a non-conflicting channel assignment and a NE is reached.

One advantage of Local-RS is that there exist no hard-to-color topologies[2]. A disadvantage of Local-RS is that it does not assign any priority to the radios, and so misses out on common optimizations such as allowing highly connected radios to select channels earlier. Because Local-RS is a best-response strategy to the potential game described in theorem 4.2.2, we know that it will converge to a NE.

Local-RS, being a strictly local algorithm, does not have the same knowledge requirements as Local-DIA, with the conflict neighborhood consisting of channel information from itself and 1-hop neighbors. Thus, there is no concept of partial knowledge. Having less than 1-hop knowledge will prevent any strategy from avoiding interference, while having more than 1-hop knowledge provides no advantage to Local-RS.

Just as with DIA and Local-DIA, the Local-RS strategy also captures aspects of learning and reasoning. After every iteration, the number of channels available in $A^{\text{cc}} \setminus \{c_j | j \in \mathcal{N}_i\}$ is non-increasing. This is because the cognitive elements learn not to choose channels that in previous iterations were the subject of interference. In a similar fashion, the radios act upon the observed interfering channels and use reasoning to select from the remaining non-conflicting channels.

[2] A hard-to-color topology means that no implementation of the algorithm can exactly color the topology.

6.3.2 Dynamic Networks

There are several possible strategies to deal with dynamic changes in the network due to the addition and removal of radios. The simplest is to simply restart Local-RS after a dynamic event, re-assigning channels. Since this is the same as the original operation of Local-RS the expected performance of this restart strategy is the same as the performance of Local-RS when run on any topology.

Another strategy is to continue utilizing Local-RS after the dynamic event. The topology of the network will have changed, changing the conflict neighborhood of some radios. Those radios that share channel assignments with their new conflict neighbors will update their channel selections according to the same rules and order of selection as used under Local-RS initially.

More complex strategies than these can be devised, incorporating such sub-strategies as trading channels with neighboring radios, restarting channel selection for subsets of the topology, or identifying potential spectrum trouble spots in the topology for alternate coloring. However, in Section 6.4.2 we will compare the restart to the continuation strategy and show that the continuation strategy is sufficient.

6.4 Results

To determine the effectiveness of the CN, we developed a simulation consisting of radios placed according to a uniform random distribution in x and y coordinates within a square 2-D map (with a density of $|N|$ radios/unit^2).

The initial topologies of the network are fully connected, meaning there exists a bi-directed path from every radio to every other radio. The initial power was chosen such that the induced network was 1-connected with 90% probability, adjusting the value for finite networks (see [149] for this formula). The network then utilizes full knowledge and Local-DIA to

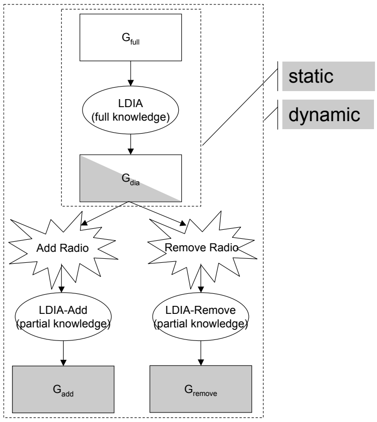


Figure 6.5: Experimental procedure for examining static and dynamic networks.

arrive at the G_{dia} topology. In the dynamic network results, a radio is then added to or removed from the network, at which point Local-DIA-Add or Local-DIA-Remove are used under various degrees of ignorance to arrive at the final topology. This experimental procedure is outlined in Figure 6.5.

6.4.1 Static Networks

First, we verify the analysis of Section 6.2.1 and show that under static, full knowledge conditions, TopoPowerControl using DIA does indeed outperform other interference avoidance schemes (specifically those described in Section 6.1.1). Figure 6.6 shows that under DIA, the average maximum connection power is 7-10% lower than other algorithms.

Next, we investigate the channel assignment performance of Local-RS. Unlike DIA, which can be proven to minimize the maximum power (see Section 6.2), the spectral optimality of Local-RS is impossible to guarantee. Determining $\kappa(U)$, the minimum number of channels required by the conflict graph, is an NP-hard problem for arbitrary graphs. However, there are well-known upper and lower bounds: $\mu(U) \leq \kappa(U)$ and $\kappa(U) \leq \Delta(U) + 1$ where $\mu(U)$ is the size of largest clique in U (a clique is a group of vertices that are fully connected) and $\Delta(U)$ is the largest degree in U. Often, this means $3 \leq \kappa(U) \leq |N|$, a rather loose bound.

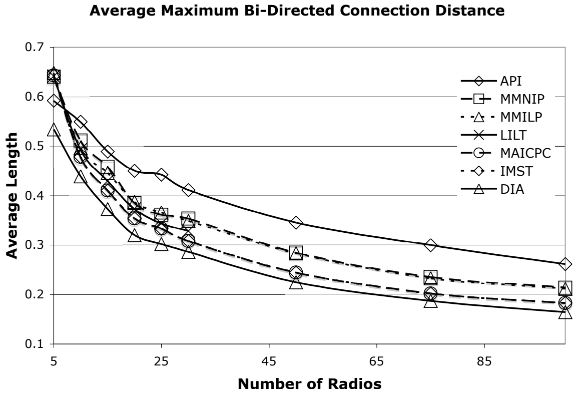


Figure 6.6: Average maximum transmission power for minimum interference algorithms.

To get a more meaningful assesment of the spectral performance, we evaluated DIA and each of the six other interference-reducing topology control algorithms mentioned in Section 6.1.1 on the same sets of scenarios ranging from 5 to 100 radios. We then transformed the resultant connectivity graphs for each algorithm to an exact distance-2 conflict graph and ran an exact coloring algorithm on it to determine the minimum number of legal channels. The exact-coloring algorithm we used, [152], implements a branch-and-bound approach, halting if it finds a solution equal to $\mu(U)$. While this algorithm can require a non-polynomial amount of time to complete, only at 100 radios did we encounter scenarios with exceptionally long termination times.

The results of this comparison are found in Figure 6.7, which shows the average minimum number of channels required by each power control algorithm. DIA has a mean channel count lower than or comparable to that of all other interference avoidance algorithms we considered, including IMST and MAICPC, which were shown in the literature to perform well. The variance of the channel count for DIA is between $5 - 13\%$ of the mean. As a comparison, IMST and MAICPC have between $5 - 12\%$ variance and the other algorithms have up to 88% variance. It is worth noting that the LILT algorithm, which compared favorably at low radio counts, does not scale well as the number of radios is increased; so we do not have results beyond the thirty-radio case.

These results indicate that, compared to other interference-reducing algorithms, DIA pro-

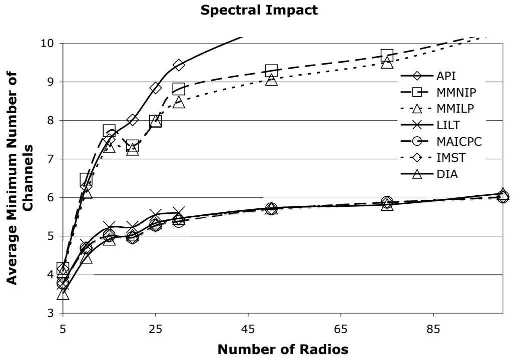


Figure 6.7: Comparison of optimal minimum average channel count for various topology control algorithms.

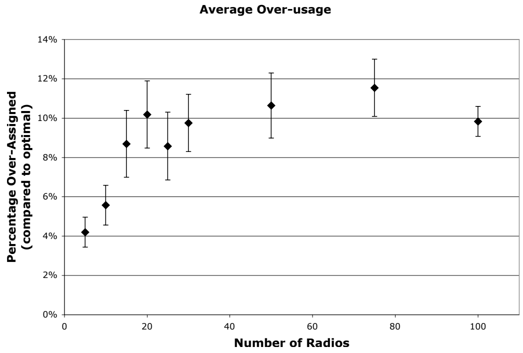


Figure 6.8: Average percentage channels overassigned (as compared to the optimal) by Local-RS, error bars represent 95% confidence intervals.

duces topologies with low conflicts. To determine the performance impact of Local-RS we compare the average additional channels required over the minimum number of channels, in Figure 6.8. This was accomplished by averaging at least 100 random permutations of π for each topology generated by DIA. The exact minimum number of channels required was determined by creating a conflict graph that represented the modified distance-2 conflict used by Local-RS. Although in the worst case Local-RS can perform arbitrarily poorly, on average the algorithm requires less than 12% additional spectrum over the optimal. This shows that although there are more complex coloring strategies possible for channel assignment, the cost of the complexity may make them poor candidates in comparison to a localized, selfish strategy such as Local-RS.

To investigate whether the random positioning of radios on the map effects the performance

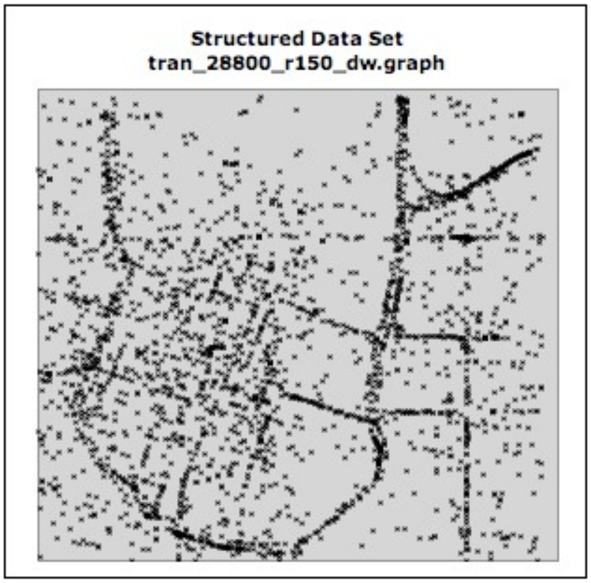


Figure 6.9: Structured handset data plotted according to (x, y) coordinates.

of Local-RS, we use a more structured distribution. The positioning data set we used comes from the Virginia Bioinformatics Institute (`http://ndssl.vbi.vt.edu/`) [153, 154, 143] and represents real-world handset locations for a time instant in Portland, Oregon. The entire data set is plotted in Figure 6.9, and the structure, particularly the presence of roads, is easily observable.

Using the average radio density for this data set, a random square cut was made inside the map of such as size that it was expected to contain 100 radios. If the cut contained fewer than 100 radios, the edges of the cut were increased in length by 10% until the cut contained at least 100 radios. If greater than 100 radios were contained, radios were randomly removed until exactly 100 remained.

The structured 100 radio data sets were given an exact conflict-free channel assignment and compared against the channel assignment from Local-RS. Whereas Local-RS operated on random data required, on average, 9.8% additional channels, in the structured data it only required, on average, 6.8% additional. Furthermore, with 95% confidence, these two averages are distinct from one another. It is not surprising that Local-RS performs slightly better on the structured data, since the structured data can utilize short, direct connections (particularly along roadways) that do not introduce as much interference. This makes the connectivity more tree-like, a graph structure that Local-RS operates particularly well on.

6.4.2 Dynamic Networks

We specifically investigate the price of ignorance for dynamic networks, knowing that under full knowledge the network can use strategies that successfully minimize both objectives. In Figure 6.10, the price of ignorance is measured for both channel and power objectives for a 50 radio network[3]. In this figure, a price of ignorance of, for instance, 1.5, means that the network objective performed 150% worse under that degree of ignorance than full knowledge.

Figure 6.10 shows that the price of ignorance is low for the maximum power objective, regardless of the amount of knowledge. This is not unexpected, as most of the time, little in the topology needs to change to incorporate a new radio. Recall from Lemma 6.2.9 that if the least power connection for a new radio is less than the current maximum power in the network, the maximum power will not increase in the new topology. Furthermore, from Lemma 6.2.10, if this least power connection is greater than the current maximum, this connection will be the new maximum. Local-DIA does not increase any radio's transmission power beyond these limits and only in very special cases will the addition of a radio reduce the maximum transmission power. Particularly as the number of radios in the network increases, the probability that adding a single radio will reduce the maximum transmission power decreases.

While the objective of minimizing the maximum transmission power is relatively unaffected by ignorance, the spectral impact objective does not fare so well. The average channel usage in Figure 6.10 was calculated using the exact coloring algorithm. The large price of ignorance at small k-hop knowledge is correlated to the fact that the total power under partial knowledge is much greater than that under full knowledge, creating more connections. In the

[3] Similar results were observed for network sizes varying from 5 to 100 radios.

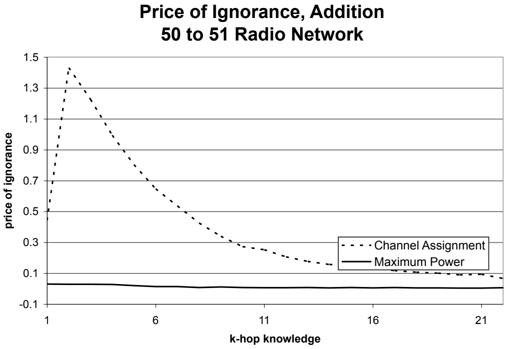


Figure 6.10: Price of ignorance for a network after a radio is added

expected sense, increasing the connectivity increases the maximum degree and clique size of the conflict graph. This in turn increases the number of channels required for conflict-free operation.

While changing the k-hop knowledge changes the connections in the topology, which in turn changes the minimum number of channels required, knowledge greater than or equal to 2-hop is full knowledge and does not change the performance of Local-RS. To determine the performance of Local-RS under dynamic changes, the restart and continuation strategies from Section 6.3.2 are compared in Figure 6.12. This shows that there is no significant difference between the two. Since the restart strategy is an acceptable baseline (with less than 12% average additional spectrum usage over the minimum), this means that the continuation strategy is also acceptable. Any other strategy that works better than the continuation strategy is an improvement on Local-RS, and should be considered as the baseline strategy. Since the continuation strategy requires less overhead than the restart strategy, it should be used.

Figure 6.11 shows the price of ignorance under radio removal for both transmission power and spectral efficiency in a 50 radio network. Unlike the results in Figure 6.10, the remove case shows that increased knowledge has a negative effect on the network objectives, performing worse under knowledge than ignorance. This surprising and counter-intuitive result can be explained by the oft heard expression *"what you don't know can't hurt you."* Under ignorance

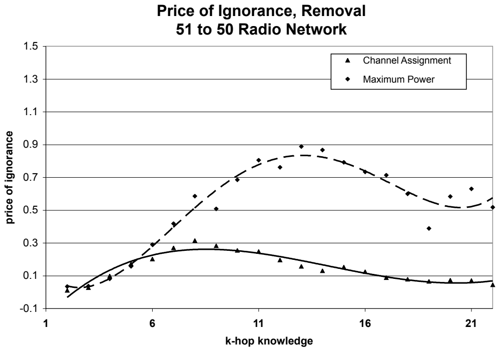


Figure 6.11: Price of ignorance for a network after a radio is removed, curve fits are within 95% confidence intervals for mean value.

(such as 3-hop knowledge), most radios are not aware that a partitioning has occurred, and as a result do not react in any way to the radio removal, causing little change to the original topology. Under larger k-hop knowledge, more radios have more k-hop neighbors that they are attempting to reconnect with, skewing the initial G_{dia} topology. As discussed earlier, Local-DIA is not a correct strategy, and particularly as the initial topology includes many additional connections over G_{dia}, it has a greater chance of converging to a sub-optimal topology. Although under ignorance the topology may be missing important low-power connections, it also does not have many extra high-power connections for Local-DIA to accidently choose from. In short, most topologies do not need much knowledge to select low-power connections for use in re-connecting the network.

The curve is non-monotonic because the of this tradeoff. For the 50 radio case depicted in Figure 6.11, leading up to 12-hop knowledge there is enough information that radios make increasingly poor choices yet not enough to begin to achieve some benefit from it. Beyond 12-hop knowledge, the performance improves because of the available knowledge. This suggests that there may exist better strategies than Local-DIA-Remove, particularly when there is larger amounts of knowledge available.

The impact on the spectral performance due to partial knowledge and dynamic changes is caused by Local-DIA rather than Local-RS. Partial knowledge has no direct effect on Local-RS, which requires only 1-hop knowledge to operate. We investigate the effect of

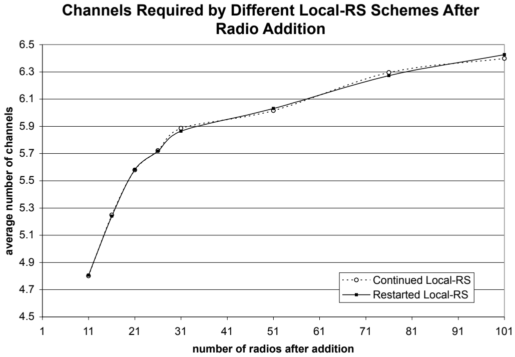


Figure 6.12: Average minimum required channels for Local-RS restart and Local-RS continuation after adding a radio

dynamic changes on Local-RS by comparing the restart and continuation strategies from Section 6.3.2 are compared in Figure 6.12. This shows that there is no significant difference between the two. Since the restart strategy is an acceptable baseline (with less than 12% average additional spectrum usage over the minimum), this means that the continuation strategy is also acceptable. Any other strategy that works better than the continuation strategy is an improvement on Local-RS, and should be considered as the baseline strategy. Since the continuation strategy requires less overhead than the restart strategy, it should be used.

6.4.3 Knowledge and Performance

The performance benefits of increasing the amount of knowledge available to the radios in the network are clear. However, this knowledge comes at a cost, requiring more transactions as the amount of knowledge increases. To understand the impact of aquiring knowledge, we need a metric that allows a comparison between the network performance (with respect to the power and spectrum objectives) and the cost of acquiring knowledge.

We have shown that DIA globally (and Local-DIA locally) minimizes the maximum transmission power in the topology. These strategies also are local minima for the sum power in the network. We have also observed that the number of channels required to have interference-free connectivity is affected by the overall connectivity of the network. As the connectivity in the network is increased, the expected number of required channels also increases. If the total power used in the network is decreased, we also expect to see improvement for both objectives.

As a proxy measurement for these objectives, we can measure the *total packet energy.* The total packet energy for data is calculated as the amount of energy required to transmit, via unicast, a data packet from every radio to every other radio, using the least-power route between every pair of radios in the network[4]. The power used by the radio at each hop along the route in the topology is summed and this value is totaled for every radio pair. To convert from power to energy, we multiply this power total by a constant equal to the length of time for a packet transmission, which assumes that all packets are of equal transmission length.

To measure the cost of maintaining the network topology, we measure the total packet energy required for knowledge by calculating the amount of energy needed to transmit an update message from every radio to each of its k-hop neighbors. As with the data measurement, this is calculated by determining the least power route from each radio to each of its k-hop neighbors. The power used by each radio to reach, via unicast, every k-hop neighbor is summed. We use the same time constant as with the data packets to convert from power to energy.

Figure 6.13 shows the total packet energy required for just data and also for the sum of the data and knowledge packets, if data packets are sent at the same frequency as knowledge updates. This shows that increasing knowledge decreases the total packet energy required for data. It also briefly decreases the total packet energy required for knowledge, but then this begins to climb. There is a "sweet spot" for energy around 5-hop knowledge, in which the sum total packet energy is lower than at full knowledge. This is the point at which the total energy cost of knowledge is minimized. To make a better sense of the amount of partial knowledge radios have, we calculated the average fraction of network a radio is aware of. For

[4] Unicast is used rather than multicast with the wireless broadcast property to give an upper bound on the amount of power required

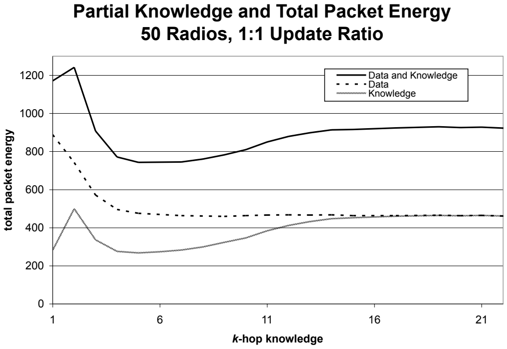


Figure 6.13: Average total packet energy required for 50 radio network

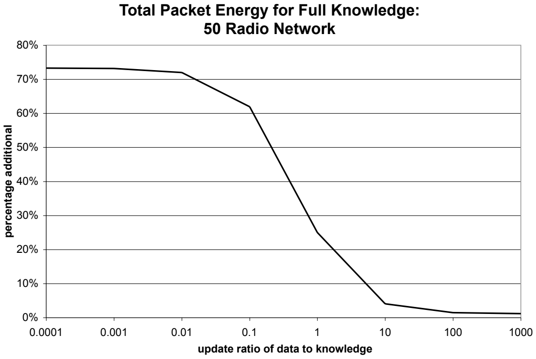


Figure 6.14: Percent additional total packet energy required under full knowledge as compared to minimum total packet energy under partial knowledge for 50 radio network

5-hop knowledge, radios have, on average, awareness of 70% of all network nodes. We also noted that this value remains same across various network sizes, with the density kept fixed. This indicates that our algorithm scales well with network size.

This example is for a low ratio of data to updates. Assuming the amount of data stays constant, as a network becomes more mobile, the number of updates required to maintain k-hop knowledge increases proportionally to the data. Figure 6.14 illustrates this sweet spot for different ratios of data to updates. It shows the percent difference between the minimum total packet energy and the full knowledge total packet energy. As expected, when the network is relatively stable, and the ratio of data to updates is high, having full knowledge gives the best performance. When the network is dynamic, and the ratio of data to updates is low, having partial knowledge gives a lower total packet energy.

6.5 Conclusion

CNs present a novel approach to achieving end-to-end objectives through learning and reasoning. By breaking down network objectives into multiple local cognitive element goals, CNs operate in a selfish, distributed and self-organized manner. The network performance these selfish cognitive elements achieve is dependent on the amount of knowledge they have about the network. While having full knowledge can allow for optimal decisions, the lack of knowledge may lead to some degree of sub-optimality. Depending on the problem to be solved and the strategy employed, sometimes having more knowledge illuminates better solutions, while other times it may just add redundancy to the system. Regardless of the network benefit these partial-knowledge solutions provide, there is always a network cost to acquiring, communicating and maintaining knowledge. Both of these factors must be taken into account to determine how much knowledge the cognitive elements need.

For the particular topology control problem examined here, the network achieves its end-to-end goals as the individual elements align their selfish objectives to the objectives of the network. We determined that this selfish approach is a potential game, meaning that network convergence and stability are guaranteed. When cognitive elements operated under full knowledge, they create a topology that minimizes the maximum transmission power and the spectral impact is comparable with other interference reduction schemes. Furthermore, the average spectrum usage of this topology is within 12% of the absolute minimum. For dynamic networks, we show that as radios join the network, more knowledge provides better spectral performance; on the contrary, when radios leave the network, some ignorance in the network achieves better performance. Finally, we determine that the cost of maintaining full knowledge is justified only when the network is fairly static.

Chapter 7

Conclusion

This research proposed the idea of a CN: a network composed of elements that, through learning and reasoning, dynamically adapt to varying network conditions in order to optimize end-to-end performance. In a CN, decisions are made to meet the requirements of the network as a whole, rather than the individual network components. We identified exciting applications of this for wireless networks, where CNs can intelligently select and adapt radio spectrum, transmission power, antenna parameters and routing tables to meet network requirements. Spectrum management and power management, in particular, have broad economic and policy implications, with interest from both the military and industry.

CNs represent the evolution of the CR concept that has swept the radio communications field by storm. Although the concept of CNs may seem an extension of CRs, it should be noted that most CR research has focused on how changes to the physical layer affect the radio. Even the fundamental question of how well multiple CRs would work together in simple network configurations has been given limited attention. By formalizing the design, architecture and tradeoffs of cognition at the network level, our work has had a broad impact in advancing the paradigm of intelligent communication devices.

7.1 Summary of Contributions

We list the problems and requirements that motivate CNs: network complexity, the wireless medium and end-to-end objectives make current approaches to network design inadequate. Complexity, particularly in wireless networks, is a problem that cannot be solved or understood easily using local and reactive networking protocols. The layered approach to the network stack at times prevents the network from achieving end-to-end goals. Although other networking concepts have addressed some of these shortcomings, none have addressed all of them.

Two of these concepts, cross-layer design and CR, are shown to be related to, but distinct from, CNs. Their strengths and weaknesses are highlighted, and many of the capabilities absent from their feature set are drawn into the definition and framework of CNs. In the course of this survey of related work, several models for analysis are presented. Additionally, a survey is made of the related work published since this research began.

Having discussed the background and related work of the CN concept, the system design and framework are formally defined. The definition of CN emphasizes its end-to-end and network-wide scope and provides the foundation for the system framework. This frameworks consists of three layers: the requirements layer, the cognitive process and the software adaptable network. The requirements layer takes the user, application or resource goals and translates them into objectives for the cognitive process. The cognitive process attempts to determine the set of parameters that meet these objectives, using learning and reasoning to make decisions. The cognitive process depends on the SAN to provide action control (over modifiable network elements) and make observations (through network status sensors).

Three critical design decisions are identified from this framework: selfishness, ignorance and control. To analyze the effect on performance that these design decisions have on the CN, the "price of a feature" metric is developed. Two extensions of this metric, the expected price and bounded price, are used extensively in the remainder of the work. The bounded price measures the worst-case performance of a design decision.

We identify two classes of CNs, the potential and quasi-concave classes, based on the selfish, rational objectives of their cognitive elements. These two classes have several important properties. When properly aligned with the network goals, they arrive at either locally (in the case of potential) or globally (in the case of quasi-concave) optimal NE. Furthermore the quasi-concave equilibrium are PONE and have interesting bounded prices under selfishness, indistinguishable information, and partial control. We identify several applications of these classes: TopoPowerControl, TopoChannelControl, PowerControl, SizeControl, and WindowControl. TopoPowerControl and TopoChannelControl are used as cognitive elements for a dual-objective CN spectrum-aware topology control problem, and a multicast variation of PowerControl is adopted in a CN multicast lifetime maximization problem.

For the multicast lifetime problem, PowerControl operates in conjunction with two other cognitive elements, DirectionControl and RoutingControl, to control the transmission power, antenna beam angle, and network routing table at every radio. The CN is designed to operate in conjunction with any routing protocol, and it was shown through simulation that it can increase the lifetime of simple multicast protocols by over 125%. Furthermore, the price of the critical design decisions shows that partial knowledge has a much larger effect on performance that either ignorance or selfishness.

Unlike the single objective multicast problem, the spectrum-aware topology control problem is has dual objectives: to simultaneously minimize both the maximum transmission power of the radios and the spectral footprint of the FDMA scheme. The DIA algorithm is proven to converge to an optimal power configuration, and the Local-RS algorithm is observed to converge to within 12% of the minimum number of frequency channels. The problem is then examined under partial knowledge for dynamic networks in which radios are added and removed. Surprisingly, it was observed that less partial knowledge is better than more partial knowledge when radios are removed from the topology. The energy cost of sharing full knowledge is shown to be high in highly dynamic networks, making partial knowledge preferable, even if it results in topologies that are less energy efficient for data.

7.2 Future Work

Given that this research was the first serious investigation into CNs, there is plenty of work yet to be done. This list represents a few open topics and questions.

- The cognitive element architecture

  While this work provided a framework for the CN to operate in, it did not specify much detail for a generic cognitive element. Functional capabilities such as a data repository, cognitive engine, or inter-element communication framework are still ill-defined and open to interpretation.

- Implementation issues

  This research focused on describing the CN concept and then implementing it via simulation. Real-world experimentations with CNs will identify design problems, flesh out implementations details, and reveal limitations. This will require the use of open radio and network platforms to act as the first SAN. Unfortunately, the existing radio hardware is still immature, and the SAN is at this point is conceptual in nature.

- Stronger learning and reasoning

  The learning and reasoning used by the cognitive elements in this work is basic, learning from well defined action sets and reasoning using a deterministic, algorithm based strategies. Future research should utilize stronger techniques, perhaps perhaps using a database to store past decisions and outcomes for learning and stochastic meta-heuristics like genetic algorithms or neural networks for reasoning. While game theory is an excellent model for selfish behaviors, it does not represent cooperative schemes well. Being able to model cooperative schemes will allow parallel reasoning schemes to be investigated.

- Distributed cognitive processes

  The cognitive processes in this work are distributed and selfish. Examining other distributed processes that are cooperative, operate well on partial knowledge, and employ techniques to avoid local solutions will give CNs a larger cognitive arsenal.

Appendix A

Acronyms

API Application Programming Interface

API Average Path Interference

ATM Asynchronous Transfer Mode

BIST Biological, Information, Social, and Technical

CA Cellular Automata

CN cognitive network

CR cognitive radio

CDPS Cooperative Distributed Problem Solving

CLICA Connected Low Interference Channel Assignment

CRM Cognitive Resource Manager

CRN Cognitive Radio Network

CSL Cognitive Specification Language

CTVR Centre for Telecommunications Value-Chain Research

DIA δ-Improvement Algorithm

DAI Distributed Artificial Intelligence

E^{2}R II End-to-End Reconfigurability Project II

EA Evolutionary Algorithm

EPF Exact Potential Function

EPG Exact Potential Game

FCC Federal Communications Commission

FDMA Frequency Division Multiple Access

FIP Finite Improvement Path

GA Genetic Algorithm

GϵPG Generalized ϵ-Potential Game

GOPG Generalized Ordinal Potential Game

HTML HyperText Markup Language

HTTP HyperText Transfer Protocol

IF Intermediate Frequency

IMST Interference Minimum Spanning Tree

IP Internet Protocol

JTRS Joint Tactical Radio System

KP Knowledge Plane

LILT Low Interference-Load Topology

KQML Knowledge Query Markup Language

MAC Medium Access Control

MAICPC Minimize the Average Interference Cost while Preserving Connectivity

MAS Multi-Agent System

MILP Mixed Integer Linear Program

MMLIP Min-Max Link Interference with property $\mathcal{P}$

MMNIP Min-Max Node Interference with property $\mathcal{P}$

MOO Multiple Objective Optimization

MST Minimum Spanning Tree

NE Nash Equilibrium

NKRL Network Knowledge Representation Language

OODA Observe, Orient, Decide and Act

OPF Ordinal Potential Function

OPG Ordinal Potential Game

OSI Open Systems Interconnection

P2P Peer-to-Peer

PHY physical

pMOOEA parallel Multiple Objective Optimization Evolutionary Algorithm

PO Pareto Optimal

PONE Pareto Optimal Nash Equilibrium

pMST power Minimum Spanning Tree

QoS Quality of Service

RF Radio Frequency

RKRL Radio Knowledge Representation Language

SAN Software Adaptable Network

SCA Software Communications Architecture

SDR Software Defined Radio

SDS Sequential Dynamic System

SGML Standard Generalized Markup Language

SINR Signal to Interference and Noise Ratio

SLA Service Layer Agreement

SON Service Overlay Network

SSH Secure Shell

SyDS Synchronous Dynamic System

TDMA Time Division Multiple Access

UMTS Universal Mobile Telecommunications System

WLAN Wireless Local Area Network

WPG Weighted Potential Game

XML eXtensible Markup Language

Bibliography

[1] J. Mitola, *Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio.* PhD thesis, Royal Institute of Technology (KTH), 2000.

[2] S. Haykin, "Cognitive radio: Brain-empowered wireless communication," *IEEE Journal on Selected Areas in Communication*, vol. 23, pp. 201–220, February 2005.

[3] R. Berezdivin, R. Breinig, and R. Topp, "Next-generation wireless communications concepts and technologies," *IEEE Communications Magazine*, vol. 40, no. 3, pp. 108–116, 2002.

[4] A. Alexiou and M. Haardt, "Smart antenna technologies for future wireless systems: Trends and challenges," *IEEE Communications Magazine*, vol. 42, no. 9, pp. 90–97, 2004.

[5] E. Gelenbe, R. Lent, and Z. Xu, "Design and performance of cognitive packet networks," *Performance Evaluation*, vol. 46, no. 2-3, pp. 155–176, 2001.

[6] B. Schwartz, A. W. Jackson, W. T. Strayer, W. Zhou, R. D. Rockwell, and C. Partridge, "Smart packets for active networks," in *OPENARCH '99*, pp. 90–97, 1999.

[7] P. Mähönen, J. Riihijärvi, M. Petrova, and Z. Shelby, "Hop-by-hop toward future mobile broadband IP," *IEEE Communications Magazine*, vol. 42, no. 3, pp. 138–146, 2004.

[8] C. Ramming, "Cognitive networks," in *DARPATech*, 2004.

[9] R. Saracco, "Forecasting the future of information technology: How to make research investment more cost-effective," *IEEE Communications Magazine*, vol. 41, pp. 38–45, December 2003.

[10] J. O. Neel, *Analysis and Design of Cognitive Radio Networks and Distributed Radio Resource Management Algorithms.* PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, September 2006.

[11] S. A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution.* Oxford University Press, 1993.

[12] K. Kelly, *Out of Control: The Rise of Neo-biological Civilization.* Addison-Wesley, 1994.

[13] F. Heylighen, "Building a science of complexity," in *Proc. of the 1988 Annual Conference of the Cybernetics Society*, 1988.

[14] C. L. Barrett, S. Eubank, and M. V. Marathe, "Modeling and simulation of large biological, information and socio-technical systems: An interaction based approach," tech. rep., Virginia Bioinformatics Institute, 2005.

[15] D. J. Watts, *Small Worlds.* Princeton Studies in Complexity, Princeton University Press, 1999.

[16] P. Wegner, "Why interaction is more powerful than algorithms," *Communications of the ACM*, vol. 40, no. 5, pp. 80–91, 1997.

[17] R. Albert and A.-L. Barabasi, "Statistical mechanics of complex networks," *Reviews of Modern Physics*, vol. 74, no. 1, p. 47, 2002.

[18] C. Barrett, A. Marathe, M. V. Marathe, and M. Drozda, "Characterizing the interaction between routing and MAC protocols in ad-hoc networks," in *Proc. of MobiHoc 2002*, pp. 92–103, 2002.

[19] C. Prehofer and C. Bettstetter, "Self-organization in communication networks: Principles and design paradigms," *IEEE Communications Magazine*, pp. 78–85, July 2005.

[20] F. Heylighen, C. Gershenson, S. Staab, G. W. Flake, D. M. Pennock, D. C. Fain, D. De Roure, K. Aberer, W.-M. Shen, O. Dousse, and P. Thiran, "Neurons, viscose fluids, freshwater polyp hydra-and self-organizing information systems," *IEEE Intelligent Systems and Their Applications*, vol. 18, no. 4, pp. 72–86, 2003.

[21] T. D. Wolf and T. Holvoet, "Emergence versus self-organisation: Different concepts but promising when combined," in *Lecture Notes in Computer Science*, vol. 3464, pp. 1–15, May 2005.

[22] M. Roth and S. Wicker, "Termite: A swarm intelligence routing algorithm for mobile wireless ad-hoc networks," *Springer SCI Series: Swarm Intelligence and Data Mining*, 2005.

[23] J. D. Vriendt, P. Lainé, C. Lerouge, and X. Xu, "Mobile network evolution: A revolution on the move," *IEEE Communications Magazine*, pp. 104–111, April 2002.

[24] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful," in *Proc. of IEEE INFOCOM 2003*, IEEE, 2003.

[25] V. Paxson and S. Floyd, "Wide area traffic: The failure of Poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 226–244, 1995.

[26] D. C. Verma, "Service level agreements on IP networks," *Proceedings of the IEEE*, vol. 92, pp. 1382–1388, September 2004.

[27] Z. Duan, Z.-L. Zhang, and Y. T. Hou, "Service overlay networks: SLAs, QoS, and bandwidth provisioning," *IEEE/ACM Transactions on Networking*, vol. 11, pp. 870–883, December 2003.

[28] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A new resource reservation protocol," *IEEE Communications Magazine*, pp. 116–127, May 2002.

[29] J. Jin and K. Nahrstedt, "QoS specification languages for distributed multimedia applications: A survey and taxonomy," *IEEE Multimedia*, vol. 11, no. 3, pp. 74–87, 2004.

[30] S. Chakrabarti and A. Mishra, "QoS issues in ad-hoc wireless networks," *IEEE Communications Magazine*, pp. 142–148, February 2001.

[31] W. Kumwilaisak, Y. T. Hou, Q. Zhang, W. Zhu, C. C. J. Kuo, and Y.-Q. Zhang, "A cross-layer quality-of-service mapping architecture for video delivery in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 10, pp. 1685–1698, 2003.

[32] J. O. Neel, R. M. Buehrer, J. H. Reed, and R. P. Gilles, "Game-theoretic analysis of a network of cognitive radios," in *Proc. of the Midwest Symposium on Circuits and Systems*, vol. 3, pp. 409–412, 2002.

[33] J. Mitola and G. Q. Maguire., "Cognitive radio: Making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, 1999.

[34] H. S. Berger, P. Kodzy, J. Hoffmeyer, and M. Shafi, "P1900 series," tech. rep., IEEE Standards, March 2006.

[35] Federal Communications Commission, "Notice of proposed rulemaking on cognitive radio," Tech. Rep. FCC 03-322, FCC, 2003.

[36] J. Boyd, "A discourse on winning and losing: Patterns of conflict," 1986.

[37] Y. Labrou and T. Finin, "A proposal for a new KQML specification," Tech. Rep. CS-97-03, University of Maryland Baltimore County, Baltimore, MD, 1997.

[38] T. Finin, J. Weber, G. Wiederhold, M. Genesereth, R. Fritzon, D. McKay, J. McGuire, R. Pelavin, S. Shapiro, and C. Beck, "Draft specification of the KQML agent-communication language," tech. rep., The DARPA Knowledge Sharing Initiative, 1993.

[39] P. Mähönen, "Cognitive trends in making: Future of networks," in *Proc. of IEEE PIMRC 2004*, vol. 2, pp. 1449–1454, 2004.

[40] R. Burgess and S. Mende, “Configuration languages: Theory and implementation,” tech. rep., E2R Consortium, 2005.

[41] V. Srivastava, J. O. Neel, A. B. Mackenzie, R. Menon, L. A. DaSilva, J. E. Hicks, J. H. Reed, and R. P. Gilles, “Using game theory to analyze wireless ad-hoc networks,” *IEEE Communications Surveys*, vol. 7, no. 4, pp. 46–56, 2005.

[42] J. E. Hicks, A. B. MacKenzie, J. O. Neel, and J. H. Reed, “A game theory perspective on interference avoidance,” in *Proc. of IEEE GLOBECOM 2004*, December 2004.

[43] V. Srivastava and M. Motani, “Cross-layer design: A survey and the road ahead,” *IEEE Communications Magazine*, vol. 43, no. 12, pp. 112–119, 2005.

[44] T. ElBatt and A. Ephremides, “Joint scheduling and power control for wireless ad-hoc networks,” in *Proc. of IEEE INFOCOM 2002*, vol. 2, pp. 976–984, 2002.

[45] M. Chiang, “To layer or not to layer: Balancing transport and physical layers in wireless multihop networks,” in *Proc. of IEEE INFOCOM 2004*, vol. 4, pp. 2525–2536, 2004.

[46] V. Kawadia and P. R. Kumar, “A cautionary perspective on cross-layer design,” *IEEE Wireless Communications*, vol. 12, no. 1, pp. 3–11, 2005.

[47] R. Winter, J. H. Schiller, N. Nikaein, and C. Bonnet, “CrossTalk: Cross-layer decision support based on global knowledge,” *IEEE Communications Magazine*, vol. 44, no. 1, pp. 93–99, 2006.

[48] V. Raisinghani and S. Iyer, “Cross-layer feedback architecture for mobile device protocol stacks,” *IEEE Communications Magazine*, vol. 44, no. 1, pp. 85–92, 2006.

[49] S. Khan, Y. Peng, E. Steinbach, M. Sgroi, and W. Kellerer, “Application-driven cross-layer optimization for video streaming over wireless networks,” *IEEE Communications Magazine*, vol. 44, no. 1, pp. 122–130, 2006.

[50] M. Gong, “An architecture for cross-layer design in wireless ad-hoc networks.” Unpublished manuscript, 2005.

[51] H. Varian, “How to build an economic model in your spare time,” in *Passion and Craft: Economists at Work* (M. Szenberg, ed.), University of Michigan Press, 1997.

[52] A. H. Bond and L. Gasser, *Readings in distributed artificial intelligence.* Morgan Kauffman Publishers, 1988.

[53] N. R. Jennings, K. Sycara, and M. Wooldridge, “A roadmap of agent research and development,” *Autonomous Agents and Multi-Agent Systems*, vol. 1, pp. 7–38, 1998.

[54] K. P. Sycara, “Multiagent systems,” *AI Magazine*, vol. 19, no. 2, pp. 79–92, 1998.

[55] T. Dietterich and P. Langley, “Machine learning for cognitive networks: Technology assessment and research challenges,” tech. rep., Oregon State University, 2003.

[56] M. J. Wooldridge, *An Introduction to Multiagent Systems.* John Wiley and Sons, 2002.

[57] T. D. Wolf and T. Holvoet, “Towards a methodology for engineering self-organising emergent systems,” in *Proc. of SOAS 2005*, 2005.

[58] G. Weiss, ed., *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence.* Cambridge, MA: MIT Press, 1999.

[59] D. H. Friend, R. W. Thomas, A. B. MacKenzie, and L. A. DaSilva, *Cognitive Networks*, ch. Distributed Learning and Reasoning in Cognitive Networks: Methods and Design Decisions. Wiley-Interscience, 2007.

[60] T. Roughgarden, *Selfish Routing and the Price of Anarchy.* The MIT Press, 2005.

[61] C. H. Papadimitriou, “Algorithms, games, and the Internet,” in *Proc. of STOC 2001*, 2001.

[62] N. Nisan and A. Ronen, “Algorithmic mechanism design (extended abstract),” in *Proc. of STOC 1999*, (New York, NY, USA), pp. 129–140, ACM Press, 1999.

[63] L. A. DaSilva, “Pricing for QoS-enabled networks: A survey,” *IEEE Communications Surveys*, pp. 2–8, Second Quarter 2000.

[64] S. Zhong, J. Chen, and Y. R. Yang, “Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks,” in *Proc. of IEEE INFOCOM 2003*, pp. 1987–1997, IEEE, April 2003.

[65] O. Ileri, S.-C. Mau, and N. B. Mandayam, “Pricing for enabling forwarding in self-configuring ad-hoc networks,” *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 151–162, January 2005.

[66] P. Michiardi and R. Molva, “Analysis of coalition formation and cooperation strategies in mobile ad-hoc networks,” *Ad Hoc Networks*, vol. 3, pp. 192–219, 2005.

[67] T. Moreton and A. Twigg, “Trading in trust, tokens, and stamps,” in *Workshop on Economics of Peer to Peer Systems*, 2003.

[68] R. J. Glauser, “Time dependent statistics of the Ising model,” *Journal of Mathematical Physics*, vol. 4, pp. 294–307, February 1963.

[69] R. A. Holley and T. M. Liggett, “Ergodic theorems for weakly interacting infinite systems and the voter model,” *The Annals of Probability*, vol. 3, pp. 643–663, August 1975.

[70] D. Griffeath, “The basic contact process,” *Stochastic Processes and their Applications*, vol. 11, no. 2, pp. 151–185, 1981.

[71] C. Asavathiratham, *The Influence Model: A Tractable Representation for the Dynamics of Networked Markov Chains*. PhD thesis, Massachusetts Institute of Technology, October 2000.

[72] T. G. Robertazzi, *Computer Networks and Systems: Queueing Theory and Performance Evaluation*. Springer-Verlag, 1994.

[73] S. Wolfram, *Cellular Automata and Complexity: Collected Papers*. Addison-Wesley, 1994.

[74] S. Wolfram, *A New Kind of Science*. Wolfram Media, 2002.

[75] C. L. Barrett and C. M. Reidys, “Elements of a theory of computer simulation I: Sequential CA over random graphs,” *Applied Mathematics and Computation*, vol. 98, no. 2-3, pp. 241–259, 1999.

[76] C. L. Barrett, H. S. Mortveita, and C. M. Reidys, “Elements of a theory of simulation II: sequential dynamical systems,” *Applied Mathematics and Computation*, vol. 107, no. 2-3, pp. 121–136, 2000.

[77] C. L. Barrett, H. S. Mortveita, and C. M. Reidys, “Elements of a theory of simulation III: equivalence of SDS,” *Applied Mathematics and Computation*, vol. 122, no. 3, pp. 325–340, 2001.

[78] R. V. Sole and S. Valverde, “Information theory of complex networks: on evolution and architectural constraints,” Tech. Rep. 03-11-061, Santa Fe Institute, 2003.

[79] J. Hasty, D. McMillen, F. Isaacs, and J. J. Collins, “Computational studies of gene regulatory networks: In numero molecular biology,” *Nature Reviews Genetics*, vol. 2, no. 4, pp. 268–279, 2001.

[80] N. Barkai and S. Leibler, “Robustness in simple biochemical networks,” *Nature*, vol. 387, pp. 913–917, 1997.

[81] D. Bray, “Protein molecules as computational elements in living cells,” *Nature*, vol. 376, no. 6538, pp. 307–312, 1995.

[82] A. Osyczka, “Design optimization,” in *Multicriteria optimization for engineering design* (J. S. Gero, ed.), pp. 193—227, Academic Press, 1985.

[83] C. A. C. Coello, “An updated survey of GA-based multiobjective optimization techniques,” *ACM Computing Surveys*, vol. 32, pp. 109–143, June 2000.

[84] N. J. Nilsson, *Artificial Intelligence: A New Synthesis.* Morgan Kauffman Publishers, 1998.

[85] E. Brookner, *Tracking and Kalman Filtering Made Easy.* Wiley-Interscience, 1998.

[86] K. S. Narendra and M. A. Thathachar, *Learning Automata: An Introduction.* Prentice Hall, 1989.

[87] C. Unsal, *Intelligent Navigation of Autonomous Vehicles in an Automated Highway System: Learning Methods and Interacting Vehicles Approach.* PhD thesis, Virginia Polyechnic Institute and State University, January 1997.

[88] M. A. Haleem and R. Chandramouli, "Adaptive downlink scheduling and rate selection: A cross-layer design," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 6, pp. 1287–1297, 2005.

[89] T. W. Rondeau, B. Li, C. J. Rieser, and C. W. Bostian, "Cognitive radios with genetic algorithms: Intelligent control of software defined radios," in *SDR Forum Technical Conference*, pp. C–3–C–8, 2004.

[90] D. A. V. Veldhuizen, J. B. Zydallis, and G. B. Lamont, "Issues in parallelizing multiobjective evolutionary algorithms for real world applications," in *SAC '02: Proc. of the 2002 ACM symposium on Applied computing*, (New York, NY, USA), pp. 595–602, ACM Press, 2002.

[91] R. W. Thomas, L. A. DaSilva, and A. B. Mackenzie, "Cognitive networks," in *Proc. of IEEE DySPAN 2005*, pp. 352–360, Nov. 2005.

[92] J. O. Neel, J. H. Reed, and R. P. Gilles, "Convergence of cognitive radio networks," in *Proc. of IEEE WCNC 2004*, vol. 4, pp. 2250–2255, 2004.

[93] G. Ganesan and Y. Li, "Cooperative spectrum sensing in cognitive radio networks," in *Proc. of IEEE DySPAN 2005*, pp. 137–143, 2005.

[94] S. M. Mishra, A. Sahai, and R. W. Brodersen, "Cooperative sensing among cognitive radios," in *Proc. of IEEE ICC 2006*, vol. 4, pp. 1658–1663, 2006.

[95] P. Pawelczak, R. V. Prasad, L. Xia, and I. G. M. M. Niemegeers, "Cognitive radio emergency networks - requirements and design," in *Proc. of IEEE DySPAN 2005*, pp. 601–606, 2005.

[96] D. Raychaudhuri, N. B. Mandayam, J. B. Evans, B. J. Ewy, S. Seshan, and P. Steenkiste, "CogNet: An architectural foundation for experimental cognitive radio networks within the future Internet," in *Proc. ACM/IEEE MobiArch'06*, pp. 11–16, Dec. 2006.

[97] D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski, “A knowledge plane for the Internet,” in *Proc. of SIGCOMM ’03*, pp. 3–10, ACM Press, 2003.

[98] D. Bourse, M. Muck, O. Simon, N. Alonistioti, K. Moessner, E. Nicollet, D. Bateman, E. Buracchini, G. Chengeleroyen, and P. Demestichas, “End-to-end reconfigurability (E^2R II): Management and control of adaptive communication systems.” Presented at IST Mobile Summit 2006, June 2006.

[99] P. Demestichas, V. Stavroulaki, D. Boscovic, A. Lee, and J. Strassner, “m@ANGEL: Autonomic management platform for seamless cognitive connectivity to the mobile internet,” *IEEE Communications Magazine*, vol. 44, no. 6, pp. 118–127, 2006.

[100] P. Sutton, L. E. Doyle, and K. E. Nolan, “A reconfigurable platform for cognitive networks,” in *Proc. of CROWNCOM 2006*, 2006.

[101] R. W. Thomas, D. H. Friend, L. A. DaSilva, and A. B. MacKenzie, “Cognitive networks: Adaptation and learning to achieve end-to-end performance objectives,” *IEEE Communications Magazine*, vol. 44, pp. 51–57, December 2006.

[102] D. Marr, *Vision: a computational investigation into the human representation and processing of visual information.* W. H. Freeman, 1982.

[103] P. N. Johnson-Laird, *The Computer and the Mind.* Harvard University Press, 1988.

[104] M. A. L. Thathachar and P. S. Sastry, *Networks of Learning Automata.* Kluwer Academic Publishers, 2004.

[105] E. J. Friedman and S. Shenker, “Learning by distributed automata.” available at http://citeseer.ist.psu.edu/friedman93learning.html, 1993.

[106] M. Katz, “Towards cooperative and cognitive wireless networks.” Application Session, IEEE ICC 2007, June 2007.

[107] T. Roughgarden and E. Tardos, “How bad is selfish routing?,” in *Proc. of the 41st Annual Symposium on the Foundations of Computer Science*, pp. 93–102, 2000.

[108] D. Monderer and L. S. Shapley, “Potential games,” *Games and Economic Behavior*, vol. 14, no. 1, pp. 124–143, 1996.

[109] S. Ramanathan, “A unified framework and algorithm for (T/F/C) DMA channel assignment in wireless networks,” in *Proc. of IEEE INFOCOM 1997*, vol. 2, pp. 900–907, 1997.

[110] R. S. Komali and A. B. MacKenzie, “Distributed topology control in ad-hoc networks: A game theoretic perspective,” in *IEEE Consumer and Comm. Networking Conf.*, vol. 1, pp. 563–568, Jan. 2006.

[111] A. Clementi, P. Penna, and R. Silvestri, "Hardness results for the power range assignment problem in packet radio networks," in *Proc. of APPROX 1999*, vol. 1671, pp. 195–208, July 1999.

[112] S. Uryasev and R. Y. Rubinstein, "On relaxation algorithms in computation of noncooperative equilibria," *IEEE Transactions on Automatic Control*, vol. 39, no. 6, pp. 1263–1267, 1994.

[113] H. Nikaido and K. Isoda, "Note on noncooperative convex games," *Pacific Journal of Mathematics*, vol. 5 Supplement 1, pp. 807–815, 1955.

[114] J. B. Krawczyk and S. Uryasev, "Relaxation algorithms to find nash equilibria with economic applications," *Environmental Modeling and Assessment*, vol. 5, pp. 63–73, 2000.

[115] D. Gabay and H. Moulin, "On the uniqueness and stability of nash-equilibria in noncooperative games," in *Applied Stochastic Control in Econometrics and Management Science* (A. Bensoussan, P. Kleindorfer, and C. S. Tapiero, eds.), vol. 130 of *Contributions to Economic Analysis*, pp. 271–293, North Holland Publishing Company, 1980.

[116] R. D. Yates, "A framework for uplink power control in cellular radio systems," *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 1341–1347, September 1995.

[117] I. L. Glicksberg, "A further generalization of the Kakutani fixed point theorem, with application to Nash equilibrium points," *Proceedings of the American Mathematical Society*, vol. 3, pp. 170–174, Feb. 1952.

[118] K. Fan, "Fixed-point and minimax theorems in locally convex topologically linear spaces," *Proceedings of the National Academy of Science*, vol. 38, pp. 121–126, 1952.

[119] F. Berggren, *Power Control, Transmission Rate Control and Scheduling in Cellular Radio Systems.* PhD thesis, Royal Institute of Technology (KTH), May 2001.

[120] T. Ylonen and C. Lonvick, "SSH communications security," RFC 4253, IETF Standards Track Working Group, 2006.

[121] Y. Yang and R. Kravets, "Achieving delay guarantees in ad-hoc networks by adapting IEEE 802.11 contention windows," in *Proc. of IEEE INFOCOM 2006*, 2006.

[122] S. K. S. Gupta and B. Wang, "Energy-efficient multicast protocols," in *Resource Management in Wireless Networking* (M. Cardei, I. Cardei, and D.-Z. Du, eds.), Kluwer Academic Publishers, 2004.

[123] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Energy-aware wireless networking with directional antennas: The case of session-based broadcasting and multicasting," *IEEE Transactions On Mobile Computing*, vol. 1, pp. 176–191, July-September 2002.

[124] K. N. Wood, *Directional Communications to Improve Multicast Lifetime in Ad Hoc Networks.* Doctoral dissertation, Virginia Polytechnic Institute and State University, Blacksburg, VA, September 2006.

[125] K. Wood and L. A. DaSilva, "Optimization of multicast routing for network lifetime with directional listening," in *Proc. of BROADNETS 2005*, pp. 494–500, October 2005.

[126] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Energy-limited wireless networking with directional antennas: The case of session-based multicasting," in *Proc. of IEEE INFOCOM 2002*, vol. 1, pp. 190–199, 2002.

[127] H. L. V. Trees, *Optimum Array Processing.* No. 4 in Detection Estimation and Modulation Theory, Wiley-Interscience, 2002.

[128] M. Gong, *Improving the Capacity in Wireless Ad-hoc Networks through Multiple Channel Operation: Design Principles and Protocols.* PhD thesis, Virginia Tech, 2005.

[129] M. Burkhart, P. von Rickenbach, R. Wattenhofer, and A. Zollinger, "Does topology control reduce interference?," in *Proc. of IEEE MobiHoc 2004*, (New York, NY, USA), pp. 9–19, ACM Press, 2004.

[130] X.-Y. Li, K. Moaveni-Nejad, W.-Z. Song, and W.-Z. Wang, "Interference-aware topology control for wireless sensor networks," in *Proc. of IEEE SECON 2005*, pp. 263–274, 2005.

[131] K.-D. Wu and W. Liao, "On constructing low interference topology in multihop wireless networks," in *Proc. of IEEE ICC 2006*, 2006.

[132] T. Johansson and L. Carr-Motyčková, "Reducing interference in ad hoc networks through topology control," in *Proc. of the 2005 Joint Work. on Found. of Mob. Comp.*, pp. 17–23, 2005.

[133] M. K. Marina and S. R. Das, "A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks," in *Proc. of BROADNETS 2005*, 2005.

[134] A. P. Subramanian, H. Gupta, and S. R. Das, "Minimum-interference channel assignment in multi-radio wireless mesh networks," in *Proc. of ICNP 2005*, November 2005.

[135] S. O. Krumke, M. V. Marathe, and S. S. Ravi, "Models and approximation algorithms for channel assignment in radio networks," *Wireless Networks*, vol. 7, no. 6, pp. 575–584, 2001.

[136] H. Balakrishnan, C. L. Barrett, V. A. Kumar, M. V. Marathe, and S. Thite, "The distance-2 matching problem and its relationship to the MAC-layer capacity of ad hoc wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 1069–1079, 2004.

[137] D. A. Grable and A. Panconesi, "Nearly optimal distributed edge colouring in o(log log n) rounds," *Random Structures and Algorithms*, vol. 10, pp. 385–405, May 1997.

[138] V. S. A. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan, "End-to-end packet-scheduling in wireless ad-hoc networks," in *SODA '04: Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, (Philadelphia, PA, USA), pp. 1021–1030, Society for Industrial and Applied Mathematics, 2004.

[139] P. Santi, "Topology control in wireless ad hoc and sensor networks," *ACM Computing Surveys (CSUR)*, vol. 37, pp. 164 – 194, March 2005.

[140] M. Bahramgiri, M. Hajiaghayi, and V. S. Mirrokni., "Fault-tolerant and 3-dimensional distributed topology control algorithms in wireless multi-hop networks," in *Proc. of ICCCN,*, pp. 392–397, 2002.

[141] N. Li, J. Hou, and L. Sha, "Design and analysis of an mst-based topology control algorithm," *IEEE Transactions on on Wireless Communications*, vol. 4, no. 3, pp. 1195–1206, 2005.

[142] D. M. Blough, M. Leoncini, G. Resta, and P. Santi, "The k-neighbors approach to interference bounded and symmetric topology control in ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 9, pp. 1267–1282, 2006.

[143] E. L. Lloyd, R. Liu, M. V. Marathe, R. Liu, R. Ramanathan, and S. S. Ravi, "Algorithmic aspects of topology control problems for ad hoc networks," *Journal of Mobile Networks and Applications*, no. 10, pp. 19–34, 2005.

[144] L. Li, P. Bahl, Y.-M. Wang, J. Y. Halpern, and R. Wattenhofer, "Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks," in *Proc. of ACM PODC 2001*, pp. 264–273, 2001.

[145] T. Melodia, D. Pompili, and I. F. Akyildiz, "Optimal local topology knowledge for energy efficient geographical routing in sensor networks," in *Proc. of IEEE INFOCOM 2004*, 2004.

[146] S. Eidenbenz, V. S. A. Kumar, and S. Zust, "Equilibria in topology control games for ad hoc networks," *Mob. Netw. Appl.*, vol. 11, no. 2, pp. 143–159, 2006.

[147] T. Moscibroda, S. Schmid, and R. Wattenhofer, "On the topologies formed by selfish peers," in *Proc. of ACM PODC 2006*, pp. 133–142, 2006.

[148] R. S. Komali, A. B. MacKenzie, and R. P. Gilles, "Effect of selfish node behavior on efficient topology design." January 2007.

[149] C. Bettstetter, "On the minimum node degree and connectivity of a wireless multihop network.," in *In Proc. ACM Intern. Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pp. 80–91, June 2002.

[150] X.-Y. Li, Y. Weng, P.-J. Wan, W.-Z. Song, and O. Frieder, "Localized low-weight graph and its applications in wireless ad hoc networks," in *Proc. of IEEE INFOCOM 2004*, vol. 1, pp. 431–442, 2004.

[151] M. Kubale, ed., *Graph Colorings*. No. 352 in Contemporary Mathematics, American Mathematical Society, english ed., 2004.

[152] M. A. Trick, "color.c." Implementation of DSATUR-based Branch and Bound in C; `http://mat.gsia.cmu.edu/COLOR/solvers/trick.c`, November 1994.

[153] K. Atkins, C. Barrett, R. Beckman, K. Bisset, M. Drozda, S. Eubank, C. Engelhart, N. Hengartner, G. Istrate, A. Kumar, M. V. Marathe, M. Morin, C. Reidys, S. Ravi, P. Romero, R. Pistone, S. Pathak, J. Smith, P. Stretz, and S. Zust, "Adhopnet: Integrated tool for end-to-end analysis of extremely large next generation communication networks, volume I and II," Tech. Rep. Technical LA-UR-03-2076, LA-UR-03-2077, Los Alamos National Laboratories, 2003.

[154] D. C. Engelhart, A. Sivasubramaniam, C. L. Barrett, M. V. Marathe, J. P. Smith, and M. Morin, "A spatial analysis of mobility models: Application to wireless ad hoc network simulation," in *Annual Simulation Symposium*, pp. 35–42, 2004.